NATIONAL URANIUM TAILINGS PROGRAM

Energy, Mines and Resources Canada
Énergie, Mines et Ressources Canada

Canada

COMPARISON OF THE PROBABILISTIC SYSTEMS ASSESSMENT CODES, UTAP AND SYVAC3, APPLIED TO THE LONG-TERM ASSESSMENT OF URANIUM MILL TAILINGS IN CANADA

DSS File #15SQ.23317-5-1720
Contract #0SQ85-00293

VOLUME I: MAIN REPORT

A Research Report Prepared for:

The National Uranium Tailings Program
Canada Centre for Mineral and Energy Technology
Energy, Mines and Resources Canada
555 Booth Street
Ottawa, Ontario
K1A 0G1

by

B.W. Goodwin and T.H. Andres
Atomic Energy of Canada Research Company
Whiteshell Nuclear Research Establishment
Pinawa, Manitoba
R0E 1L0

1986 June

49pp.

MICROMEDIA

ABSTRACT

This report describes a comparison of the output of two codes: UTAP (Uranium Tailings Assessment Program) and SYVAC3 (Systems Variability Analysis Code, generation 3). Both codes are probabilistic systems assessment codes, written to serve as tools in the preparation of long-term environmental assessments. Both codes consist of a central core of 'executive' modules, to which are linked one or more submodels that represent the physical system being assessed.

UTAP has been used in a preliminary environmental assessment of the impact of a generic uranium mine tailings site.[1] Available documentation for this assessment includes a description of the model representing the tailings site. From these written descriptions, a set of model-related subroutines were written and linked with SYVAC3. SYVAC3 was then run, using the supplied data, to generate suitable output for comparison with the output from UTAP. This report provides details on the steps used to prepare the code for SYVAC3, and a description of the comparison of the results from UTAP and SYVAC3

NOTICE

This report was prepared by Atomic Energy of Canada Limited (AECL) under contract to the National Uranium Tailings Program (NUTP). Neither the NUTP nor AECL makes any warranty, expressed or implied, or assumes any liability with respect to any damage or loss incurred as a result of the use made of the information in this publication.

1/ SENES. 1985. Probabilistic Model Development for the Assessment of the Long-term Effects of Uranium Mill Tailings in Canada - Phase II. Volume 1 - Main Report. Volume 2 - Workshop Notes and Topical Discussion Papers. Volume 3 - Probabilistic Analysis User's Manual and Documentation for the Uranium Tailings Assessment Program (UTAP)

TABLE OF CONTENTS

VOLUME I	PAGE
ABSTRACT	i
NOTICE	i
LIST OF FIGURES	iii
LIST OF TABLES	iii
EXECUTIVE SUMMARY	1
1.0 INTRODUCTION	2
1.1 Background	2
1.2 Scope of Work	3
1.3 Quality Assurance Procedures	4
2.0 DEVELOPMENT OF THE TAILINGS MODEL CODE	6
2.1 Overview of the Tailings Model	6
2.2 Overview of SYVAC3	13
2.3 Specifications and Data Dictionary	15
2.4 Interfaces to SYVAC3	16
2.5 Design of the Tailings Model for Use in SYVAC3	18
2.6 Coding and Testing	21
3.0 COMPARISON OF UTAP AND SYVAC3	29
3.1 Comparison Criteria	29
3.2 Comparison of Deterministic Results	30
3.3 Comparison of Probabilistic Results	34
3.4 Summary of the Comparisons	34
4.0 SUMMARY AND CONCLUSIONS	38
ACKNOWLEDGMENTS	42
REFERENCES	42

VOLUME II

APPENDIX A. ANCILLARY DOCUMENTATION FROM SENES	A-1
APPENDIX B. SPECIFICATIONS FOR THE TAILINGS MODEL	B-1
APPENDIX C. DATA DICTIONARY FOR THE TAILINGS MODEL	C-1
APPENDIX D. RECORDS OF THE QUALITY ASSURANCE WALKTHROUGHS FOR THE SPECIFICATIONS	D-1
APPENDIX E. SAMPLE OUTPUT FROM SOFTOOL'S PROGRAMMING ENVIRONMENT AUDITOR	E-1
APPENDIX F. SAMPLE RESULTS FROM STRUCTURED BASIS MODULE TESTING AND FUNCTIONAL TESTING	F-1
APPENDIX G. DETAILED OUTPUT FROM SYVAC3	G-1
APPENDIX H. LISTINGS OF CODE DEVELOPED FOR THE URANIUM TAILINGS MODEL AS IMPLEMENTED IN SYVAC3	H-1

LIST OF FIGURES

		Page
FIGURE 1	Conceptual Structure of the Uranium Tailings Model	6
FIGURE 2	Schematic Overview of the SYVAC Assessment Approach	14
FIGURE 3	Detailed Data Flow Diagrams for Time-Independent Calculations	19
FIGURE 4	Detailed Data Flow Diagrams for Time-Dependent Calculations	20
FIGURE 5	Structure Chart for Subroutine SIMLAT, the Main Interface between SYVAC3 and the Uranium Tailings Model	22
FIGURE 6	Structure Chart for Subroutine SIMMIX	23
FIGURE 7	Structure Chart for Subroutine SIMUNZ	24
FIGURE 8	Structure Chart for Subroutine SIMRAN	25
FIGURE 9	Structure Chart for Subroutine SIMWAT	26
FIGURE 10	Structure Chart for Subroutine SIMAIR	27
FIGURE 11	Plots of the Dose Variables as a Function of Time	36

LIST OF TABLES

TABLE 1	First Comparison of SYVAC3 and UTAP Results for the Deterministic Run	31
TABLE 2	Final Comparison of SYVAC3 and UTAP Results for the Deterministic Run	33
TABLE 3	Comparison of SYVAC3 and UTAP Statistics for the Probabilistic Run	35
TABLE 4	Summary of Major Types of Deficiencies in the Documentation of the Uranium Tailings Model	39

EXECUTIVE SUMMARY

This report describes a comparison of two probabilistic assessment codes applied to an assessment of a hypothetical uranium tailings system. The two codes are the Uranium Tailings Assessment Program (UTAP) and the third generation of the SYstems Variability Analysis Code (SYVAC3).

UTAP was developed by SENES consultants, under contract to the National Uranium Tailings Program. SENES also applied UTAP in a preliminary study of a uranium tailings system (SENES, 1985). SYVAC3 was developed by Atomic Energy of Canada Limited, as a long-term assessment tool to study nuclear fuel waste disposal. Previous generations of SYVAC3 had been applied to several interim studies of waste disposal concepts.

To perform the comparison of UTAP and SYVAC3, the documentation of the component submodels in UTAP (SENES, 1985) was used to develop equivalent submodels for SYVAC3. SYVAC3 was successfully modified to provide an assessment of a uranium tailings system. The development of code for SYVAC3 followed a set of quality assurance procedures described in detail in this report. During the development of the modifications, some problems with the reference documentation of UTAP were noted. Details are given in this report of the problems found and the steps undertaken to address them.

The comparison of UTAP and SYVAC3 involved both a single deterministic simulation, and a set of 500 simulations for use in a probabilistic analysis. In both comparisons, the final SYVAC3 results were in qualitative agreement with the UTAP results. Some quantitative agreement was also demonstrated, but there still exists some quantitative differences between the two codes for some of the results. The final SYVAC3 results for the deterministic simulation were produced following several cycles of comparison, investigation of differences, and modification of the specifications and code for the uranium tailings model used in SYVAC3. The quality of these (deterministic) comparisons was improved substantially between the preparation of the first and the final SYVAC3 results. An exhaustive examination of all remaining differences was not achieved, however, due to a shortage in the time available for this study.

Two major conclusions were reached: The first is that detailed comparisons of this sort make a valuable contribution to the quality assurance of large and complicated codes. The work described here was successful in this regard, since it does point out where improvements are required so that better quality assurance of UTAP may be realized. The second is that a considerable amount of effort must be given to the specification of the system model, particularily when the system model is very complicated. It may not be possible to achieve a comprehensive system model specification using only written text; other documentation techniques, such as minispecifications, are probably essential. This is particularily true when it is desirable to produce a complicated code that must meet detailed quality assurance guidelines, or when such a code is to be used in a detailed code intercomparison such as the intercomparison described in this report.

1.0 INTRODUCTION

1.1 Background

One of the goals of the National Uranium Tailings Program (NUTP) is to investigate and develop procedures for the long-term environmental assessment of the disposal of uranium mill and mine tailings (Harrison et al., 1985). The probabilistic systems assessment approach was chosen for development, since this approach can address the uncertainty and variability inherent in a disposal system that will operate far into the future.

The probabilistic assessment approach relies heavily on numerical simulations to produce system performance predictions. Mathematical models are developed of all processes that could significantly influence the behaviour of the disposal system and its potential impacts on man. Uncertainty and variability of the parameters in the mathematical models are accounted for by describing them in terms of suitable probability density functions, representing the range of feasible values that the parameters may have over the period of interest.

The mathematical complexity of the models and data usually requires the use of advanced numerical algorithms implemented on a digital computer. Validation and verification are very important in the development of these models and the associated computer codes. In this report, validation means checking the implementation of a model on a computer against real world observations. It is a global process to ensure that model predictions generated by computer are relevant to the real world. Verification is the process undertaken at each stage in the development of the computer code to prevent the introduction of error. Verification is a local process to check that, at each stage, the new product (e.g., code design, data structure design, coded subroutine) is an accurate representation of a product at an earlier stage. For example, verification steps ensure that specifications match the conceptual model, and that code matches specifications. This usage of terms is not universal, but it has been adopted by numerous authors (e.g., Dunn and Ullman, 1982; Deutsch, 1982), and it conforms to usage by the International Atomic Energy Agency (IAEA, 1982). In addition, this report uses the term "code quality assurance" to denote the collection of verification steps used to ensure that the final computer code accurately represents the specified models and data.

In the strictest sense, it is not possible to validate a probabilistic assessment model that is meant to predict impacts far into the future, since these impacts cannot be observed to establish the accuracy of the model. (It may be possible to perform a partial validation, for example, a suitable natural analog has been operating for some time, and the model is used successfully to predict the current state of that system from a knowledge of the system at earlier times). The question of validating the uranium tailings model is not part of the scope of this project.

Verification of a probabilistic assessment model is both possible and practical, and many known techniques can be applied. For example, keeping a comprehensive 'audit trail' of the code development and providing supporting documentation from detailed research models that are driven by data from a research and development program help to verify stages in the development process. Quality assuring a code by comparing it with established codes that perform the same or similar functions is a another reliable approach of particular value for codes that are very complicated logically or numerically. The uranium tailings model requires such a complicated code, and the main objective of this report is to describe such a comparison. Another objective is to demonstrate quality assurance procedures to show that the code developed is an accurate representation of the uranium tailings model.

The two codes being compared are the Uranium Tailings Assessment Program (UTAP) and generation 3 of the SYstems Variability Analysis Code (SYVAC3). Both codes are probabilistic systems assessment codes, consisting of a set of executive models linked with a set of system-specific submodels. UTAP, for example, consists of the RANSIM module and the PAC module (SENES, 1985). RANSIM is an executive module, which uses PAC to interface with a model specifically developed for a generic uranium tailings system (SENES, 1985).

SYVAC3 has been developed by Atomic Energy of Canada Limited (AECL) for the long-term assessment of the disposal of nuclear fuel waste (SYVAC3 was preceded by two earlier versions: SYVAC1 (Dormuth and Sherman, 1981) and SYVAC2 (Sherman et al., 1986)). The executive modules in SYVAC3 consist of about 10 000 lines of code, which have been developed using an extensive set of quality assurance procedures. The executive modules in SYVAC3 were developed so that the code could be readily adapted to assessments other than nuclear fuel waste disposal. In this report, SYVAC3 is linked with the tailings model originally developed for UTAP (SENES, 1985).

1.2 Scope of Work

The main objective of this study was to compare simulation results from UTAP with simulation results from a version of SYVAC3 that has been applied to a uranium tailings disposal facility. This comparison would lead to an independent check of UTAP and contribute to its quality assurance. The project to achieve this objective was divided into several tasks:

(1) Specify the uranium tailings model for use with SYVAC3. The detailed specifications are drawn from the documentation given in Appendices A to F inclusive of Volume 3 of the SENES report (SENES, 1985), together with other information supplied by SENES.

(2) From the specifications, design and code the uranium tailings model to link with SYVAC3, following the quality assurance procedures described in section 1.3.

(3) Run a set of random simulations using input data from the SENES report (SENES, 1985). Results from SYVAC3 were compared with those from UTAP; specifically the comparisons involved the results in Figures 4.2 and 4.3 in Volume 1 of the SENES report (SENES, 1985). In addition, one of the recommendations of an

early specification walkthrough was that it would be helpful to compare results from a single deterministic simulation (that is, a simulation in which all required input parameters have known values). SENES consultants forwarded a copy of the detailed input and output required for such a comparison.

(4) Compile all the documentation of the development process and the analysis of comparison results into this report.

To support these tasks, a number of conversations took place with SENES consultants. Appendix A contains a record of the ancillary information that was acquired in this way.

1.3 Quality Assurance Procedures

Much of the time spent on this study was used to develop a representation of the uranium tailings model that could easily be linked with the SYVAC3 executive modules. During this development, it was important to ensure that the code developed was an accurate representation of the documented tailings model, so that the comparison of UTAP and SYVAC3 results would be based on an identical component model. To meet this requirement, a set of quality assurance procedures was applied. (Most of the SYVAC3 executive modules had previously been subjected to a similar set of quality assurance procedures.) These procedures included the following:

(1) A formal and comprehensive set of specifications were prepared for all the new code developed for use with SYVAC3. The requirements for the code were taken from the SENES report (SENES, 1985), together with ancillary information provided by SENES (see Appendix A). The specifications for the uranium tailings model used with SYVAC3 are reproduced in Appendix B, and details are provided showing how the model interacts with the SYVAC3 executive modules. Appendix C contains a copy of the data dictionary, which is an integral component of the set of specifications.

(2) Formal structured walkthroughs were held for the specifications. Appendix D contains a record of the walkthroughs for the specification phase of the project.

(3) Softool's "Programming Environment" (PE) (Softool, 1984) was demonstrated for some of the code developed for the uranium tailings model used in SYVAC3. PE is an integrated methodology and supporting software designed to help in the development of high-quality computer software. The PE "auditor" code, for example, offers a number of features that greatly augment the checking performed by standard FORTRAN compilers. Appendix E contains an example of the application of the PE "auditor" code to successive versions of one of the subroutines (BISECT) developed for the uranium tailings model.

(4) Structured basis module testing and functional testing were performed on one of the subroutines (BISECT), which was used in several places in the uranium tailings model in SYVAC3 to find roots of equations.

(5) Special tests were considered for the code developed. For the most part, the submodels within the uranium tailings model are relatively simple, and no special tests were required. A major exception is the source term component of the tailings model, which is very complicated, and which has no known parallel. In this case, extensive manual checking of intermediate results was performed for the deterministic simulation.

(6) The entire UTAP model, as specified in Appendix B, was implemented fully in the programming language APL, an interactive programming language with powerful mathematical operators. (A model in APL can be implemented with about 10% of the code of a FORTRAN implementation, and it can be completed correspondingly more quickly.) The APL implementation runs only one simulation at a time, and it does not sample parameters, but instead inputs their values interactively. It is also much slower than the FORTRAN implementation. Nevertheless, the use of APL gave two important benefits at minimal cost. First of all, it provided an independent test of the FORTRAN implementation, because the two implementations were independently coded (in different computer languages) from the same specifications, and should yield the same results. Comparison of these two implementations became the primary method of testing the FORTRAN implementation. The second benefit of the APL implementation is that it enabled rapid modification of the model when differences between SYVAC3 and UTAP results were being investigated. These two benefits were obtained at the cost of about 15 hours of code development time.

2.0 DEVELOPMENT OF THE TAILINGS MODEL CODE

2.1 Overview of the Tailings Model

The uranium tailings model used here was developed by SENES consultants, and has been documented in detail (SENES, 1985). The following description is a condensed summary of the model.

Figure 1 shows the conceptual structure of the uranium tailings model. It consists of 10 major component modules, where module is defined here to mean one or more related processes required to produce one or more major calculated variables. The uppermost module, TSOURC, produces calculated variables required by lower levels of modules (LAKE, RADON, DUST and LAKEPH), which in turn may produce variables required by the next level of modules, and so forth. The lowest level modules, DOSE and LAKEPH, produce the final major variables of interest, which include the total annual average effective whole body dose equivalent commitment[1] to a 'critical' man, and the pH of the lake.

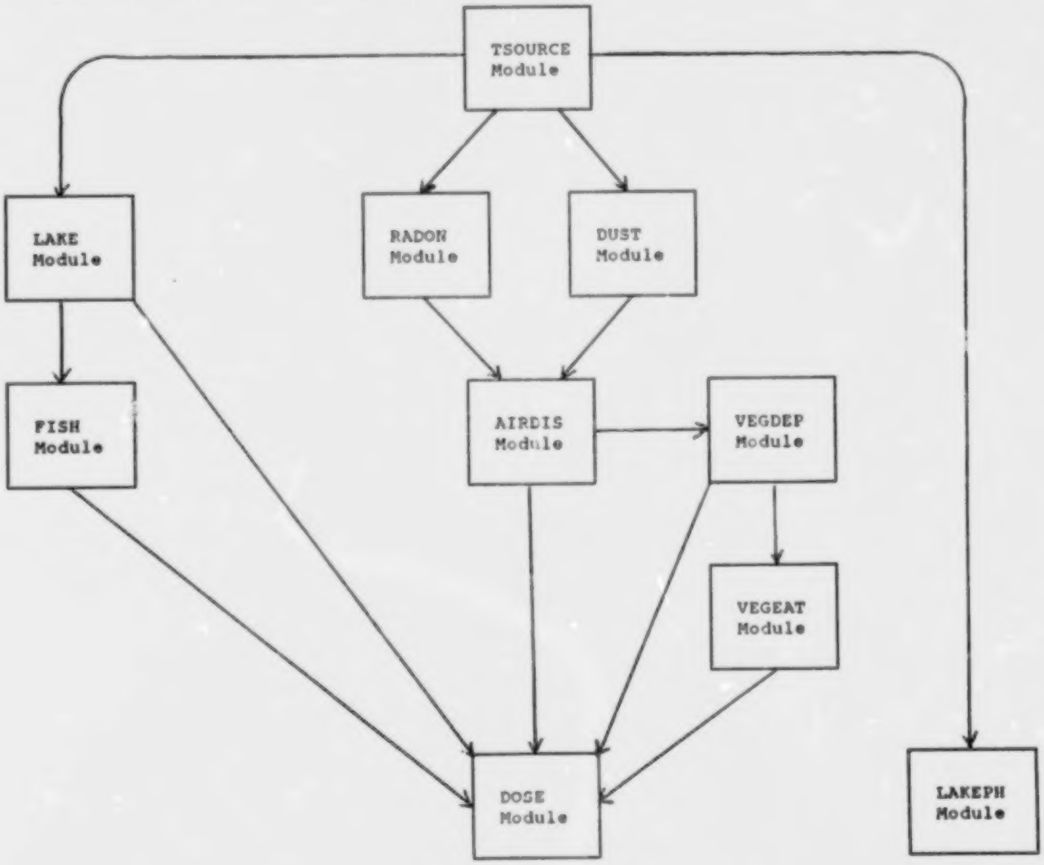

TSOURCE Module
LAKE Module
RADON Module
DUST Module
FISH Module
AIRDIS Module
VEGDEP Module
VEGEAT Module
DOSE Module
LAKEPH Module

Figure 1. Conceptual Structure of the Uranium Tailings Model (adapted from SENES, 1985)

---

[1] 1/ referred to as dose hereinafter

Most of the documentation for these modules is found in the Appendices of Volume 3 of the SENES report (SENES, 1985). Some additional information is reproduced in Appendix A of this report. A brief description of the function of each of these modules is given below.

TSOURC: The tailings 'source term' module. TSOURC calculates the waterborne and airbornes source term variables that are required by the LAKE, RADON, DUST and LAKEPH modules. TSOURC is by far the most complicated module. It represents the evolution in time of the tailings site, and includes an unsaturated zone and a saturated zone. Processes that are modelled include the following (SENES, 1985):

- Oxidation of pyrite within the unsaturated zone. It is assumed that all pyrite within the unsaturated zone is exposed to oxygen from the atmosphere. As the pyrite is oxidized, it produces sulphate ions, which migrate downward towards the saturated zone.

- Dissolution of gypsum within the upper layer of the unsaturated zone. The constant influx of fresh water from rainfall and melting snow gradually dissolves the gypsum in the unsaturated zone. At the start of the simulation, it is assumed that the entire unsaturated zone contains a uniform distribution of gypsum. As time progresses, the gypsum in the upper layer dissolves and moves towards the saturated zone. The upper layer is also referred to as the gypsum-depleted layer (gypsum-free may be more appropriate).

- Precipitation of gypsum within the lower layer of the unsaturated zone. Dissolved gypsum from the upper layer of the unsaturated zone moves down into the lower layer of the unsaturated zone, along with sulphate ions from the oxidation of pyrite. The presence of these 'excess' sulphate ions causes some gypsum (a hydrated calcium sulphate) to reprecipitate. The lower layer is also referred to as the gypsum-enriched layer.

- Movement of the front between the upper and lower layers of the unsaturated zone. At the start of the simulation, there is no 'upper' layer. As the gypsum at the top of the unsaturated zone dissolves and moves downwards, the upper (gypsum-depleted) layer is gradually formed. The remaining lower portion of the unsaturated zone, the lower (gypsum-enriched) layer, is gradually destroyed. After some time, the front reaches the interface between the unsaturated and saturated zone. At this time, only the upper (gypsum-depleted) layer remains.

- Partitioning of radium-226, associated with gypsum, between the aqueous phase and a solid phase (gypsum) within the unsaturated zone. Radium is assumed to co-precipitate with gypsum, and the concentrations of radium-226 within the unsaturated zone is affected by the dissolution and precipitation of gypsum.

- Partitioning of radium-226 from the decay of thorium-230 between the aqueous phase and solid phases (thorium ores) within the unsaturated zone. The radioactive decay of thorium-230 produces radium-226 which may be trapped within the ThO_2 matrix, or which may be released to the aqueous phase.

- Partitioning of thorium between the aqueous and solid phases. The uranium tailings initially contain thorium-232 and thorium-230 which slowly dissolve and/or decay. Concentrations of thorium-230 within the unsaturated zone are of particular interest, because this isotope decays to radium-226.

- Changing concentrations in the mixing zone. The mixing zone is defined to be the upper portion of the saturated zone, and extends from the (time-invariant) water table downwards. Contaminants are assumed to enter the mixing zone from the overlying unsaturated zone. These contaminants then exit the mixing zone, and are assumed to enter a nearby lake through a surface water pathway (movement to deep groundwaters is not modelled within this version of the SENES uranium tailings model, although the movement of deep groundwaters is treated separately in Appendix G (SENES, 1985)). The contaminants of interest are calcium, sulphate and radium-226 ions.

The TSOURC module produces seven major variables that are required by other modules. These variables, and the modules requiring them, are as follows:

CAMIX - Concentration of calcium ions in the aqueous phase of the mixing zone, as a function of time. Required by the LAKEPH module.

DEPTH1 - Depth of the upper (gypsum-depleted) layer of the unsaturated zone, as a function of time. Required by the RADON module.

DEPTH2 - Depth of the lower (gypsum-enriched) layer of the unsaturated zone, as a function of time. Required by the RADON module.

RACON1 - Concentration of radium-226 in the solid phase in the upper (gypsum-depleted) layer of the unsaturated zone, as a function of time. Required by the DUST and RADON modules.

RACON2 - Concentration of radium-226 in the solid phase in the lower (gypsum-enriched) layer of the unsaturated zone, as a function of time. Required by the RADON module.

RAMIX - Concentration of radium-226 in the aqueous phase of the mixing zone, as a function of time. Required by the LAKE module.

SO4MIX - Concentration of sulphate ions in the aqueous phase of the mixing zone, as a function of time. Required by the LAKEPH module.

LAKE: The lake module. LAKE calculates changes in concentrations of Ra-226 within the lake and lake sediments as a function of time. Processes that are modelled include the following (SENES, 1985):

- Inflow and outflow of water. It is assumed that surface runoff from the tailings area, contaminated with Ra-226, enters the lake. The lake also receives uncontaminated water from the remainder of its catchment area. It is assumed that equal volumes of water enter and exit the lake, so that its volume does not change with time.

- Changing concentrations of radium-226 in the lake water and lake sediments. The radium-226 entering the lake can be flushed out with water exiting the lake, it can be retained temporarily in the lake sediment, or it can be permanently buried in deep lake sediments.

The LAKE module requires the variable RAMIX from the TSOURC module, and produces only one variable that is required elsewhere:

CLAKE - Average annual concentration of radium-226 in the lake water, as a function of time. Required by the FISH and DOSE modules.

RADON: The radon release module. RADON calculates the release of gaseous Rn-222 from the tailings pile to the air immediately above. It is assumed that radon is released only from the unsaturated zone. Processes that are modelled in RADON include the following (SENES, 1985):

- Emission of radon-222 from the upper (gypsum-depleted) layer of the unsaturated zone. The flux of radon-222 is a function of the depth of the upper layer and the concentration of radium-226 within the upper layer.

- Emission of radon-222 from the lower (gypsum-enriched) layer of the unsaturated zone. This flux of radon-222 is a function of the depth of the lower layer and the concentration of radium-226 within the lower layer.

The RADON module requires the variables DEPTH1, DEPTH2, RACON1 and RACON2 from the TSOURC module, and produces only one variable that is required elsewhere:

EMRATE(RN) - Emission rate of radon-222 gas from the tailings unsaturated zone (upper and lower layers) as a function of time. Required by the AIRDIS module.

DUST: The dust emission module. DUST models the saltation of the tailings pile, and calculates the release, of radium-226 contained within dust particles, to the air immediately above the tailings pile. Processes that are modelled in DUST include the following (SENES, 1985):

- Release of dust particles to the air. This is due to wind erosion, and is a function of parameters such as the particle size and mean wind speed.

- Changes to the rate of emission of radium-226 to the air above the tailings pile. Radium-226 emission rates are assumed to be equal to the emission rates of dust particles released from the tailings pile multiplied by the concentration of radium-226 in the superficial layer of the unsaturated zone.

The DUST module requires the variable RACON1 from the TSOURC module, and produces only one variable that is required elsewhere:

> EMRATE(RA) - Emission rate of radium-226 in dust from the tailings surface, as a function of time. Required by the AIRDIS module.

FISH: The module for contamination of fish. FISH calculates the concentration of radium-226 in the flesh of fish in a lake. It includes the following process (SENES, 1985):

> - Bioaccumulation of radium-226 in fish, due to the exposure of fish to lake water contaminated with radium-226 A simple model is used, in which radium-226 concentrations in fish are given by the radium-226 concentrations in the lake water multiplied by an appropriate transfer coefficient.

The FISH module requires the variable RAMIX from the TSOURC module, and it produces only one variable that is required elsewhere:

> CFISH - Average annual concentration of radium-226 in edible portions of aquatic animals (fish). Required by the DOSE module.

AIRDIS: The module for dispersion of contaminants in air. AIRDIS calculates the atmospheric transport of radon gas and radium in dust particles from the tailings pile. Radon and radium in dust can be inhaled by a 'critical man' or radium-contaminated dust can fall on vegetation. The processes modelled in AIRDIS include the following (SENES, 1985):

> - Dispersion of contaminants in air by diffusion and advection. A sector-averaged Gaussian plume model is used.
>
> - Changes in the concentrations of radon-222 and radium-226 in air at a receptor. These concentrations use the Gaussian plume model and the emission rates of radon gas and radium in dust at the tailings pile.

AIRDIS requires the variables EMRATE(RN) from the RADON module, and EMRATE(RA) from the DUST MODULE, and it produces two variables that are required elsewhere:

> CAIR(RA) - Average annual concentration of radium-226 associated with dust in air at two receptors. The receptors are a 'critical man' and locally grown vegetation. Required by the DOSE module, where the receptor is a 'critical man', and by the VEGDEP module, where the receptor is locally grown vegetation.
>
> CAIR(RN) - Average annual concentration of radon-222 gas in air at a single receptor, which is a 'critical man'. Required by the DOSE module.

VEGDEP: The module for contamination of vegetation. VEGDEP calculates the concentration of radium-226 in four types of locally grown vegetation:

lichens, grass, above-ground vegetables and below-ground vegetables. The first two types of vegetation are assumed to be forage for caribou, while the second two types are assumed to be food stuff eaten by a 'critical man'. Processes that are modelled in VEGDEP include the following (SENES, 1985):

- Direct foliar retention of radium-226. It is assumed that the four types of vegetation can accumulate radium-226 that has been deposited from dust in the air onto their exposed surfaces.

- Changing concentrations of radium-226 in the edible portion of vegetation. This process takes into account parameters such as the yield density of the vegetation and the fraction of deposited radium-226 that is retained on the edible portion of the vegetation.

VEGDEP requires the variable CAIR(RA) from the AIRDIS module, and it produces one variable that is required elsewhere:

CVEG(v) - Average annual concentration of radium-226 in the edible portion of vegetation (types v=1 to v=4) due to foliar retention of radium-226 deposited from dust in the air. Vegetation types 1 and 2 are lichens and grass, respectively, and are required by the VEGEAT module. Vegetation types 3 and 4 are above-ground and below-ground vegetables, respectively, and are required by the DOSE module.

VEGEAT: The module for contamination of caribou. VEGEAT calculates the concentration of radium-226 in caribou meat, assuming that caribou ingest some of the contaminated lichens and grass (vegetation types 1 and 2). Processes modelled in VEGEAT include the following (SENES, 1985):

- Bioaccumulation of radium-226 in caribou meat, due to the exposure of caribou to lichens and grass contaminated with radium-226. A simple model is used, in which radium-226 concentrations in caribou meat are given by the radium-226 concentrations in the ingested vegetation multiplied by an appropriate transfer coefficient, an average annual feed ingestion rate, and the fraction of annual intake that is contaminated.

The VEGEAT module requires the variables CVEG(1) and CVEG(2) from the VEGDEP module, and it produces only one variable that is required elsewhere:

CMEAT - Average annual concentration of radium-226 in caribou meat, due to the ingestion of lichen and grass (vegetation types 1 and 2) that have been contaminated by the deposition of dust containing radium-226. Required by the DOSE module.

DOSE: The dose module. DOSE calculates the radiological impact to a 'critical man' due to the presence of a nearby uranium tailings pile. Processes modelled in DOSE include the following (SENES, 1985):

- Dose due to the ingestion of radium-226 in contaminated vegetation, caribou meat, drinking water from a lake, and fish.

- Dose due to the inhalation of dust containing radium-226 and the inhalation of radon-222 and its daughters.

- Total dose as a function of time.

DOSE requires CVEG(3) and CVEG(4) from the VEGDEP module, CMEAT from the VEGEAT module, CLAKE from the LAKE module, CFISH from the FISH module, and CAIR(RA) and CAIR(RN) from the AIRDIS module. It produces a number of variables that are major output variables from the uranium tailings model:

VEGDOS(v) - Annual dose due to the ingestion of radium-226 in above-ground (v=3) and below-ground (v=4) vegetables.

XMEATD - Annual dose due to the ingestion of radium-226 in caribou meat.

WATDOS - Annual dose due to the ingestion of radium-226 in drinking water from a lake.

FISHDS - Annual dose due to the ingestion of radium-226 in fish meat.

XINHIN - Annual dose due to the inhalation of radium-226 in dust.

RNDOS - Annual dose due to the inhalation of radon gas and its radioactive daughters.

TDOSE - Total annual dose due to the ingestion and inhalation of radium-226 and inhalation of radon-222

LAKEPH: The tailings lake pH module. LAKEPH calculates the pH of the lake as a function of time, due to the inflow of calcium and sulphate ions from the tailings mixing zone. Processes modelled in LAKEPH include the following (SENES, 1985):

- Dissolution equilibria of ferric hydroxide and of gibbsite. These solid phases are assumed to be in dissolution equilibrium in the lake. Other aqueous species are assumed to enter the lake at normal 'background' levels.

- Changes in concentration of calcium and sulphate ions in the lake. Calcium and sulphate concentrations in the lake are affected by two sources. The first source is their normal (time-invariant) 'background' levels, and the second source is the mixing zone of the tailings pile.

- Changes in the pH of the lake. The lake pH is affected by the different equilibrium reactions assumed to occur, and by the time-varying concentrations of calcium and sulphate ions entering the lake.

LAKEPH requires the variables CAMIX and SO4MIX from the TSOURC module. It produces only one variable that is a major output variable from the uranium tailings model:

PHLAKE - pH of the lake as a function of time.

2.2 Overview of SYVAC3

The mandate of the Canadian Nuclear Fuel Waste Management (NFWM) Program is to investigate and develop the technology for the disposal of nuclear fuel wastes in Canada (Rummery and Rosinger, 1984). One of the objectives of this program is to develop the technology required to perform a long-term post-closure assessment of different concepts for nuclear fuel waste disposal. This requirement has led to the development of a methodology based on systems variability analysis (Dormuth and Quick, 1981), which is implemented in the computer code, SYVAC.

The SYstems Variability Analysis Code (SYVAC) was designed to serve as a tool for conducting long-term environmental assessments. SYVAC provides an overall "systems" approach, since there is often a complex interplay between processes operating in a natural system, and misleading estimates of impact can be obtained when such processes are considered separately. The principal feature of SYVAC, however, is that it has been designed to accommodate variability and uncertainty in the system. This feature is important for several reasons:

- Some processes may be poorly understood, but must be quantified as well as possible.
- The assessment must be extrapolated to very long times, since radiological impacts may be potentially significant for hundreds of thousands of years.
- The concept assessment phase must incorporate data and models representative of many possible sites, since this phase of the NFWM Program is generic in nature.

The SYVAC approach is illustrated in Figure 2. Information from the research program, detailed analysis and expert opinion are used to develop simplified models and data that represent the system. For the Canadian NFWM Program, the disposal system has been divided into submodels for the vault, geosphere and biosphere. Uncertainty and variability in the data are accommodated by the use of probability distributions to define parameter values, rather than a single "best estimate" or "worst case" value. SYVAC selects a simulation defined to be a possible state of the system, by randomly sampling a value for each parameter from its specified distribution. This set of parameter values is then used within the submodels to estimate the transport of contaminants from the vault to the biosphere, and leads to a predicted impact for that simulation. The selection and prediction process is repeated, typically a thousand times or more, and the results are summarized in, for example, a histogram of predicted consequence versus their frequency of occurrence.

In this manner, SYVAC produces a distribution of estimates of impact that reflects a measure of the uncertainties associated with the predicted impacts.

SYVAC has been developed in a staged fashion, to facilitate the investigation of this approach to the study of long-term environmental assessments. The first generation of SYVAC, SYVAC1 (Dormuth and Sherman,

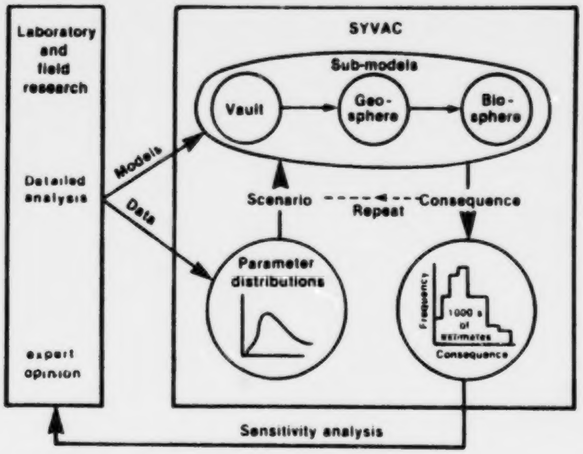

Laboratory and field research
Detailed analysis
expert opinion
Models
Data
SYVAC
Sub-models
Vault
Geo-sphere
Bio-sphere
Scenario
Repeat
Consequence
Parameter distributions
Frequency
1000 s of estimates
Consequence
Sensitivity analysis

Figure 2: Schematic Overview of the SYVAC Assessment Approach

1981), has been applied to preliminary assessments of high-level nuclear fuel waste disposal in crystalline rocks (Wuschke et al., 1981) and in the seabed (Wuschke et al., 1983), and to a preliminary assessment of intermediate-level reactor waste disposal in a shallow burial site (Guvanasen, 1985).

Further development has led to the second generation, SYVAC2 (Sherman et al., 1986). SYVAC2 has been applied recently to a second interim assessment of high-level nuclear fuel waste disposal (Wuschke et al., 1985), and to an assessment of chemical toxicity impacts associated with high-level nuclear fuel waste disposal (Goodwin et al., 1986). The basic concepts in SYVAC2 are identical to those in SYVAC1, with the major differences being in the submodels used to represent the vault, geosphere and biosphere. The development of these improved submodels was the focus of most of the effort associated with the preparation of SYVAC2, and was required to assimilate the growing body of information available from the research and development programs.

Work is now underway to develop SYVAC3. For this generation of the code, considerable effort is being devoted, once again, to develop improved submodels. It was recognized, however, that changes were also required to the SYVAC executive modules, principally to expedite the substitution of new submodels into the overall code (Goodwin, 1984). Development of SYVAC3 began in the fall of 1984, and at the time this report was written, the new SYVAC3 executive modules were complete and available for use. The majority of these modules have been subjected to a full set of quality assurance procedures (G.R. Sherman, personal communication). The SYVAC3 executive is largely a completely new code, with a number of major improvements, including the following (Andres, 1985, Goodwin and Rosinger, 1985):

-A simplified method for introducing new models and/or submodels. Section 2.4 describes the interfaces required by SYVAC3 to introduce the uranium tailings model. Andres (see Murray et al., 1986) has demonstrated the ease of use of this interface, by implementing a new model into SYVAC3 within a few hours.

-An improved method for handling convolution integrals. An occasional problem with SYVAC2 was that it could give rise to large errors in numerical integrations. Work on this problem led to a new method for handling time series in general. The time series management modules (Andres, 1985) can be used to control the simulation of a model (or a series of submodels) called by SYVAC3, and to provide, automatically, a guaranteed accuracy in the results. The time series management modules provide other functions, such as numerical integration of convolution integrals, addition of several time series and an algorithm that solves problems described by linear ordinary differential equations (Andres, 1985).

The flexibility of SYVAC3 has been demonstrated by unpublished work for the Canadian Nuclear Fuel Waste Management Program, and in a contract report for the National Uranium Tailings Program (Murray et al., 1986). Other applications of SYVAC3 are also planned or in progress. To distinguish between these different applications, the acronym for SYVAC3 has been slightly modified. SYVAC3-C, for example, is the version of SYVAC3 containing submodels required to assess the Canadian concept for nuclear fuel waste disposal. For the studies described in this report, the full designation of SYVAC3 is known as SYVAC3-T, where the "T" denotes the application to the uranium tailings assessment.

2.3 Specifications and Data Dictionary

An essential component of the quality assurance of a computer code is the specification of the requirements of the code. Specifications are used to demonstrate that a systems analysis has been carried out, and that the requirements desired can be achieved. Ideally, the complete set of specifications can be handed over to a designer and programmer, who should be able to produce the desired code without referring to any other related material. Development of suitable specifications is a time-consuming process, but there are positive returns, since the time spent on specifications means less time is required for the programming stage, and since a better quality product is produced.

The specifications for SYVAC3 were already available (G.R. Sherman, K.J. Hoffman and others, work in progress), and suitable specifications had to be developed for the uranium tailings model. The uranium tailings model was largely documented in Appendices A to F of Volume 3 of the SENES report (SENES, 1985), and this material served as the starting point for the preparation of specifications. During the first attempt at producing specifications, and also at subsequent stages, it was found that some additional information was required. The required information was provided by B.E. Halbert and M. Murray of SENES consultants, and is reproduced in Appendix A.

The final set of specifications for the uranium tailings model are presented in Appendices B and C. Appendix B contains the set of data flow diagrams, data dependency diagrams and module interface specifications. Appendix C contains the data dictionary, another important product of the quality assurance process. Several interim versions of these specifications were prepared before the final product was produced. The interim versions were subjected to formal walkthroughs by experts in different disciplines, to insure that the specifications developed accurately reflect the uranium tailings model. These walkthroughs were successful in detecting many errors in the specifications before they had a deleterious impact on the programming stage. The walkthroughs were also useful in producing a self-contained and consistent set of specifications that were suitably customized to match the requirements of the project. A full record of the walkthroughs of the specifications are reproduced in Appendix D.

2.4 Interfaces to SYVAC3

The interfaces of any system model to SYVAC3 have been designed to follow a standardized set of rules. These interfaces, along with the specifications of the model, determine the design of the model code.

There are four considerations to interface a system model to SYVAC3. These four considerations, and the major standardized rules that are in effect, are described below.

(1) INPUT FILE: A SYVAC3 input file must have the following entries:

- Title.
- Simulation control information, such as the number of simulations required.
- Time series constants (see discussion below of time-dependent calculations).
- Output options, such as whether lists of parameter values are to be generated for each simulation, and whether time series calculations are to be traced.
- Common block list. "Include" files containing common blocks are read by SYVAC3 to automatically store input parameter values into the variable locations used within the system model (see also consideration (2)).
- Parameter data, including variable names used within the system model, a brief description of the variable, the data required to define the parameter probability distribution, and the parameter units.
- Nuclide chain list (not used in this project).

The major effort in creating the input file for a particular system model is that of preparing a list of all of the data required for the input parameters and the specification of all required consequence variables. These data include information that is a part of the quality assurance procedure, such as a short parameter description and the SI units, along with the essential information, such as the probability distribution characteristics (where applicable). The uranium tailings system model had 219 parameters and consequence variables, including the named constants.

(2) COMMON BLOCKS: All system model parameter and consequence values used within SYVAC3 are stored in named COMMON blocks. These COMMON blocks must be organized and defined in "include" files, so called because they are named in INCLUDE statements within the SYVAC3 executive code. To limit interconnections within the code, and to facilitate quality assurance, the code is usually

organized in such a way that self-contained sections include only a few COMMON blocks that access only those parameter values that they require. Model COMMON blocks are referenced by three subroutines in the SYVAC3 executive (one to initialize all consequence values, one to sample and store parameter values, and one to write information out to the output files). These three SYVAC3 subroutines must be modified whenever a new model is connected to the SYVAC3 executive, to ensure that each system model COMMON block is correctly accessed. As noted above, the names of the include files are also listed in the input file, so that the SYVAC3 executive can automatically store the input parameter data into the variable locations used by the system model. The uranium tailings model used 17 COMMON blocks in 14 include files to store all required variables.

(3) TIME-INDEPENDENT CALCULATIONS: A distinction is made in SYVAC3 between time-independent and time-dependent variables. All sampled parameters are currently assumed to be constant over time (although they may define time-dependent processes). Some calculated values are also constant with time, such as the INIT. GYPSUM CONTENT IN TAILINGS (GYPBLK) and TIME TO DEPLETE GYPSUM IN UNSAT. ZONE (TFINAL). In SYVAC3, all calculated time-independent variables may be calculated in a subroutine called DEPPAR, which is executed once for every simulation. DEPPAR, which may be a dummy subroutine, must be provided when interfacing a system model to SYVAC3. It was used in the uranium tailings model to calculate 15 time-independent variables.

(4) TIME-DEPENDENT CALCULATIONS: SYVAC3 uses time series to store time-dependent variables. That is, they are stored as lists of the ordered pairs: (time, function value). The SYVAC3 executive includes an extensive set of subroutines to manipulate time series. For example, time series can be added, subtracted, multiplied, divided, convoluted, and multiplied by scalars. Each operation is carried out by a single subroutine call. These routines can be used to select variable-length time steps automatically, so as to achieve a specified target fractional error in all manipulations of the times series. The minimum and maximum number of time steps and the target fractional error are specified in the input file, as part of the 'time series constants' input. The use of the time series management routines in SYVAC3 is optional. When used, however, references to the routines must meet specified coding guidelines.

The sequence of time-dependent calculations is programmed in a SYVAC3 subroutine called SIMLAT. SIMLAT, like DEPPAR, is an interface routine that must be present (SIMLAT could also be a dummy subroutine, if necessary). SIMLAT for the uranium tailings model called six other subroutines to carry out calculations related to (1) mixing zone concentrations and lake pH, (2) unsaturated zone concentrations, (3) emission rates of Ra and Rn, (4) doses by the air pathways, (5) doses by the water pathways, and (6) total dose. The reason for this subdivision is discussed in the following section.

2.5 Design of the Tailings Model for Use in SYVAC3

The design of the uranium tailings model implemented in SYVAC3 was determined by the specifications and the interfaces to SYVAC3. The following paragraphs describe the design phase of the system model.

The first step of the design phase involved a review of the detailed data flow diagrams. These detailed diagrams were condensed onto two pages, as shown in Figures 3 and 4. The numbered bubbles in these figures refer to the numbered data flow diagrams, which are reproduced in Appendix B.

Figure 3 shows the data flow for time-independent calculations. The subroutine DEPPAR was coded directly from this diagram and the corresponding minispecifications (from Appendix B).

Figure 4 shows the flow of time-dependent variables in the model. This figure was used to break up the complicated overall model into several submodels, as shown by the solid lines in the diagram. The advantage of this subdivision of the model is that coupling of the submodels (the number of flow lines from one to another) was kept to a minimum. One result of this subdivision is that there are differences between the original conceptual structure (see Figure 1) and the design structure of the uranium tailings model in SYVAC3. This result is reflected in part by the organization of the bubbles in Figure 4; for example, bubbles 1.2.3.1 and 4.3.4 are included within the same subdivision, whereas bubbles 1.2.3.1 and 1.2.4.2 are in different subdivisions.

Once the model had been subdivided, the minispecifications for each bubble were examined to determine the processes involved, and how these processes transformed the input data to give the output data. In many cases, two or more bubbles could be joined together as one transformation, indicated in Figure 4 by dashed lines.

A common transformation used was provided by the time series management module called CONVRT, which applies a user-defined conversion to one or more incoming time series to create one outgoing time series. For example, both bubbles 1.2.3.4 and 1.2.3.5 (in the bottom left corner of Figure 4) were implemented using CONVRT.

Bubble 1.2.3.4 had one input (ION) and one output (GYPLFT). The transformation was a finite difference solution to an ordinary differential equation. The finite difference calculations were carried out in a subroutine named ZGYPLF, which was called indirectly by CONVRT. In this example, the user-defined conversion is coded in ZGYPLF, and the transformation of the time series is provided by CONVRT.

Similarly, bubble 1.2.3.5 was implemented using the CONVRT subroutine. It had four inputs (CONCS8, ION, GYPLFT, and CONCCA) and one output (SO4MIX). In this case, the calculations were a simple combination of tests and additions. They were carried out in the subroutine ZSO4MI, which was indirectly called by CONVRT.

The subroutines ZGYPLF and ZSO4MI are examples of "user subroutines", which have a standard calling sequence and usage in SYVAC3. All

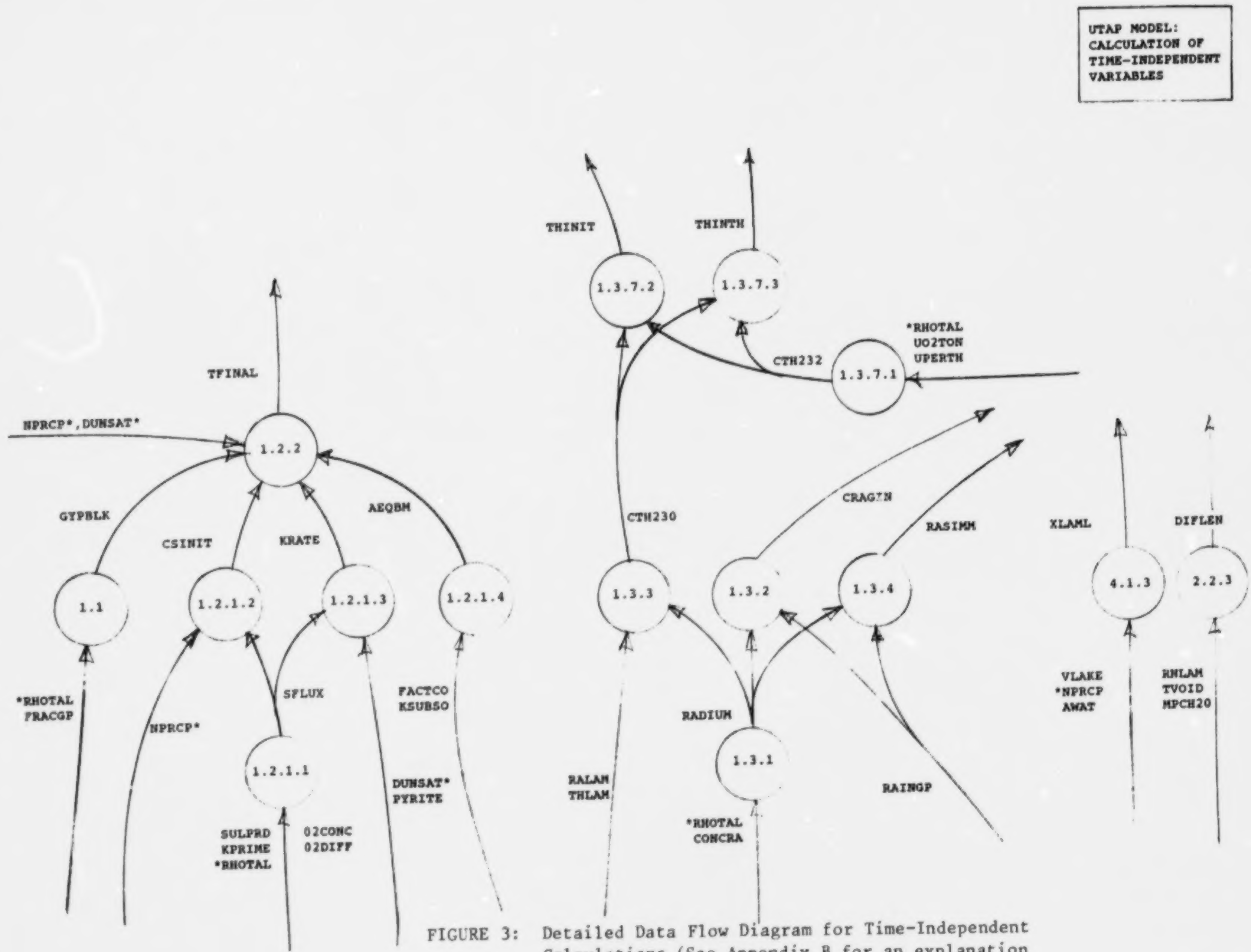

UTAP MODEL:
CALCULATION OF
TIME-INDEPENDENT
VARIABLES
TFINAL
NPRCP*,DUNSAT*
1.2.2
GYPBLK
CSINIT
KRATE
AEQBM
1.1
1.2.1.2
1.2.1.3
1.2.1.4
*RHOTAL
FRACGP
NPRCP*
SFLUX
FACTCO
KSUBSO
1.2.1.1
DUNSAT*
PYRITE
SULPRD
KPRIME
*RHOTAL
02CONC
02DIFF
THINIT
THINTH
1.3.7.2
1.3.7.3
*RHOTAL
UO2TON
UPERTH
CTH232
1.3.7.1
CTH230
CRAGTN
RASIMM
XLAML
DIFLEN
1.3.3
1.3.2
1.3.4
4.1.3
2.2.3
RADIUM
1.3.1
RALAM
THLAM
*RHOTAL
CONCRA
RAINGP
VLAKE
*NPRCP
AWAT
RNLAM
TVOID
MPCH20

FIGURE 3: Detailed Data Flow Diagram for Time-Independent Calculations (See Appendix B for an explanation of data flow diagrams.)

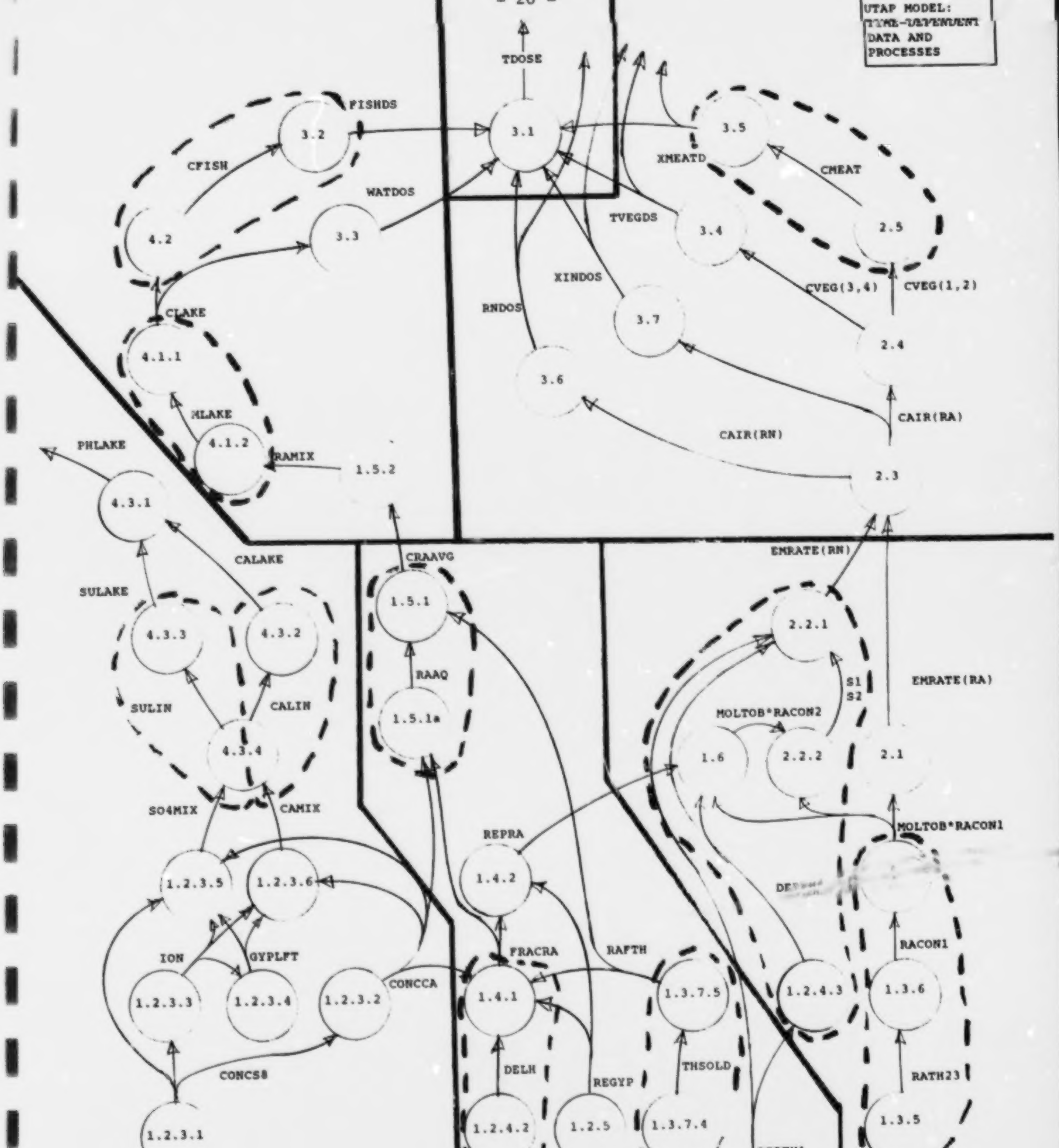

UTAP MODEL:
TIME-DEPENDENT
DATA AND
PROCESSES
TDOSE
3.1
FISHDS
3.2
CFISH
WATDOS
3.3
4.2
3.5
XMEATD
CMEAT
TVEGDS
3.4
2.5
XINDOS
CVEG(3,4)
CVEG(1,2)
RNDOS
3.7
2.4
3.6
CAIR(RA)
CAIR(RN)
2.3
CLAKE
4.1.1
MLAKE
4.1.2
RAMIX
1.5.2
PHLAKE
4.3.1
CALAKE
SULAKE
4.3.3
4.3.2
SULIN
CALIN
4.3.4
SO4MIX
CAMIX
CRAAVG
1.5.1
RAAQ
1.5.1a
EMRATE(RN)
2.2.1
S1
S2
EMRATE(RA)
MOLTOB*RACON2
1.6
2.2.2
2.1
MOLTOB*RACON1
REPRA
1.4.2
DEPTH2
1.2.3.5
1.2.3.6
ION
GYPLFT
FRACRA
RAFTH
RACON1
1.2.3.3
1.2.3.4
1.2.3.2
CONCCA
1.4.1
1.3.7.5
1.2.4.3
1.3.6
CONCS8
DELH
REGYP
THSOLD
RATH23
1.2.3.1
1.2.4.2
1.2.5
1.3.7.4
DEPTH1
1.3.5
1.2.4.1

FIGURE 4: Detailed Data Flow Diagram for Time-Dependent Calculations (See Appendix B for an explanation of data flow diagrams.)

such subroutines developed for the uranium tailings model in SYVAC3 are readily identified, since all use a subroutine name with a "Z" prefix.

Such subroutines are used with several of the time series operations. Another common time series module called SCAMUL, for example, multiplies a time series by a scalar. SCAMUL was particularly important in dose calculations. For example, bubbles 3.5, 3.6, and 3.7 were implemented as calls to SCAMUL.

In summary, the following procedure was used in the design phase. The model was divided into submodels as shown in Figures 3 and 4. All the bubbles in Figure 4 were examined one by one and converted to the appropriate transformation using time series routines, and all bubbles in Figure 3 were calculated in DEPPAR. The design of the model became a series of subroutine calls, all of which were linked to the SYVAC3 executive either directly (as in DEPPAR) or through the time-dependent SIMLAT subroutine. These calls were organized into subroutines corresponding to the submodels. The final structure is shown in the structure charts in Figures 5 to 10.

2.6 Coding and Testing

For the most part, implementation of the uranium tailings model in SYVAC3 was a straightforward procedure. About 15 man-days of work were required to code and test the SYVAC3 version of UTAP to the point where results could be compared with UTAP results.

During the development, there were some difficulties to overcome. First, because no test data were available, it had to be generated. Second, some numerical analysis was required in implementing and testing the numerical calculations. Third, some errors were discovered in the SYVAC3 executive itself, in particular in some of the time series routines that were still undergoing quality assurance testing.

A major challenge in linking the uranium tailings model with SYVAC3 was the generation of test data to ensure that the model was implemented correctly according to the specifications. Detailed results were available for one deterministic run from the original UTAP code, but these results could not be used in the development of the SYVAC3 version without weakening the later comparison of code results (that is, it might be argued that the adjusting of the tailings model implemented in SYVAC3 might be a form of 'curve fitting'). We decided, therefore, to implement the model in two separate and distinct fashions, and to compare results between the two implementations. This would be followed by a comparison of the SYVAC3 results with the UTAP results.

The first implementation of the tailings model was that for SYVAC3, in FORTRAN. It was to be used for all comparisons with UTAP results. The second implementation was carried out in the programming language, APL, on a different microcomputer. The APL version of the model was to be used only in checking the FORTRAN version of the model (in SYVAC3). Both implementations were to be developed to meet the specifications and then tested against each other until their results were in good agreement. (This comparison was greatly aided by a feature of the SYVAC3 code: the ability to trace in detail all time series calculations.)

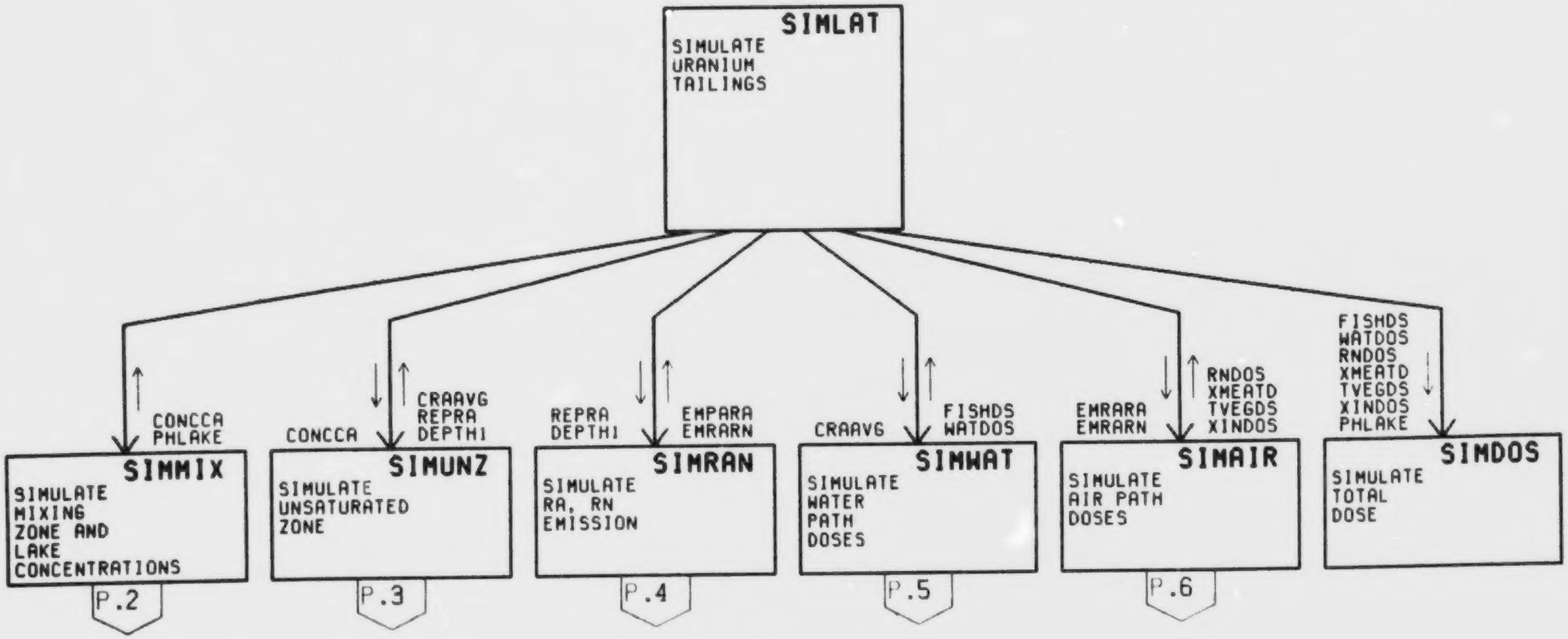

SIMLAT
SIMULATE URANIUM TAILINGS
CONCCA PHLAKE
SIMMIX
SIMULATE MIXING ZONE AND LAKE CONCENTRATIONS
P.2
CONCCA
CRAAVG REPRA DEPTH1
SIMUNZ
SIMULATE UNSATURATED ZONE
P.3
REPRA DEPTH1
EMPARA EMRARN
SIMRAN
SIMULATE RA, RN EMISSION
P.4
CRAAVG
FISHDS WATDOS
SIMWAT
SIMULATE WATER PATH DOSES
P.5
EMRARA EMRARN
RNDOS XMEATD TVEGDS XINDOS
SIMAIR
SIMULATE AIR PATH DOSES
P.6
FISHDS WATDOS RNDOS XMEATD TVEGDS XINDOS PHLAKE
SIMDOS
SIMULATE TOTAL DOSE

FIGURE 5: Structure Chart for Subroutine SIMLAT, the Main Interface between SYVAC3 and the Uranium Tailings Model. The pentagonal boxes at the bottoms of the first five boxes show that further levels of structure charts are required. SIMMIX, for example, is on page 2 of these structure charts, and is reproduced in Figure 6 of this report. Figures 7 to 10 contain the other lower level structure charts.

T. ANDRES 86-APR-02 **P.1**

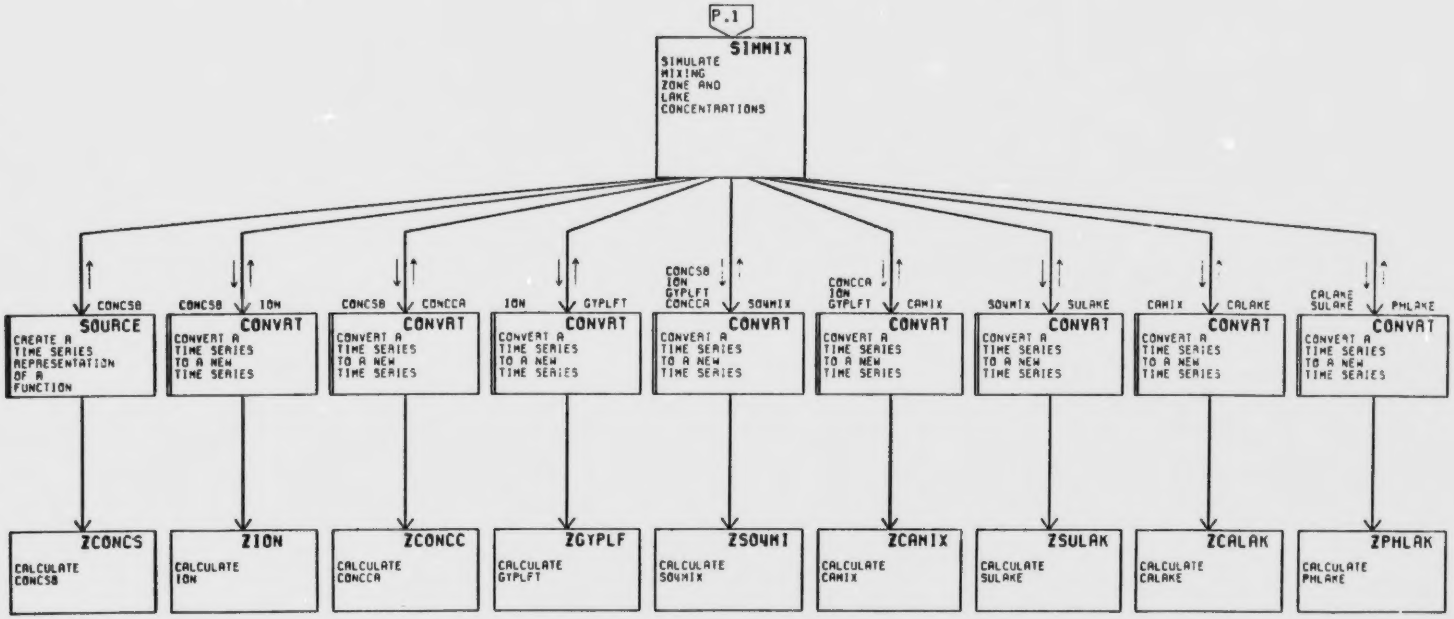

P.1
SIMMIX
SIMULATE MIXING ZONE AND LAKE CONCENTRATIONS
CONCS8
SOURCE
CREATE A TIME SERIES REPRESENTATION OF A FUNCTION
CONCS8
ION
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
CONCS8
CONCCA
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
ION
GYPLFT
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
CONCS8
ION
GYPLFT
CONCCA
SO4MIX
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
CONCCA
ION
GYPLFT
CAMIX
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
SO4MIX
SULAKE
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
CAMIX
CALAKE
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
CALAKE
SULAKE
PHLAKE
CONVRT
CONVERT A TIME SERIES TO A NEW TIME SERIES
ZCONCS
CALCULATE CONCS8
ZION
CALCULATE ION
ZCONCC
CALCULATE CONCCA
ZGYPLF
CALCULATE GYPLFT
ZSO4MI
CALCULATE SO4MIX
ZCAMIX
CALCULATE CAMIX
ZSULAK
CALCULATE SULAKE
ZCALAK
CALCULATE CALAKE
ZPHLAK
CALCULATE PHLAKE

FIGURE 6: Structure Chart for Subroutine SIMMIX. The pentagonal box at the top of the figure shows that this structure chart is referred to by a higher level of structure charts; in this case it is structure chart 1, which is reproduced in Figure 5.

T. ANDRES 86-APR-02

P.2

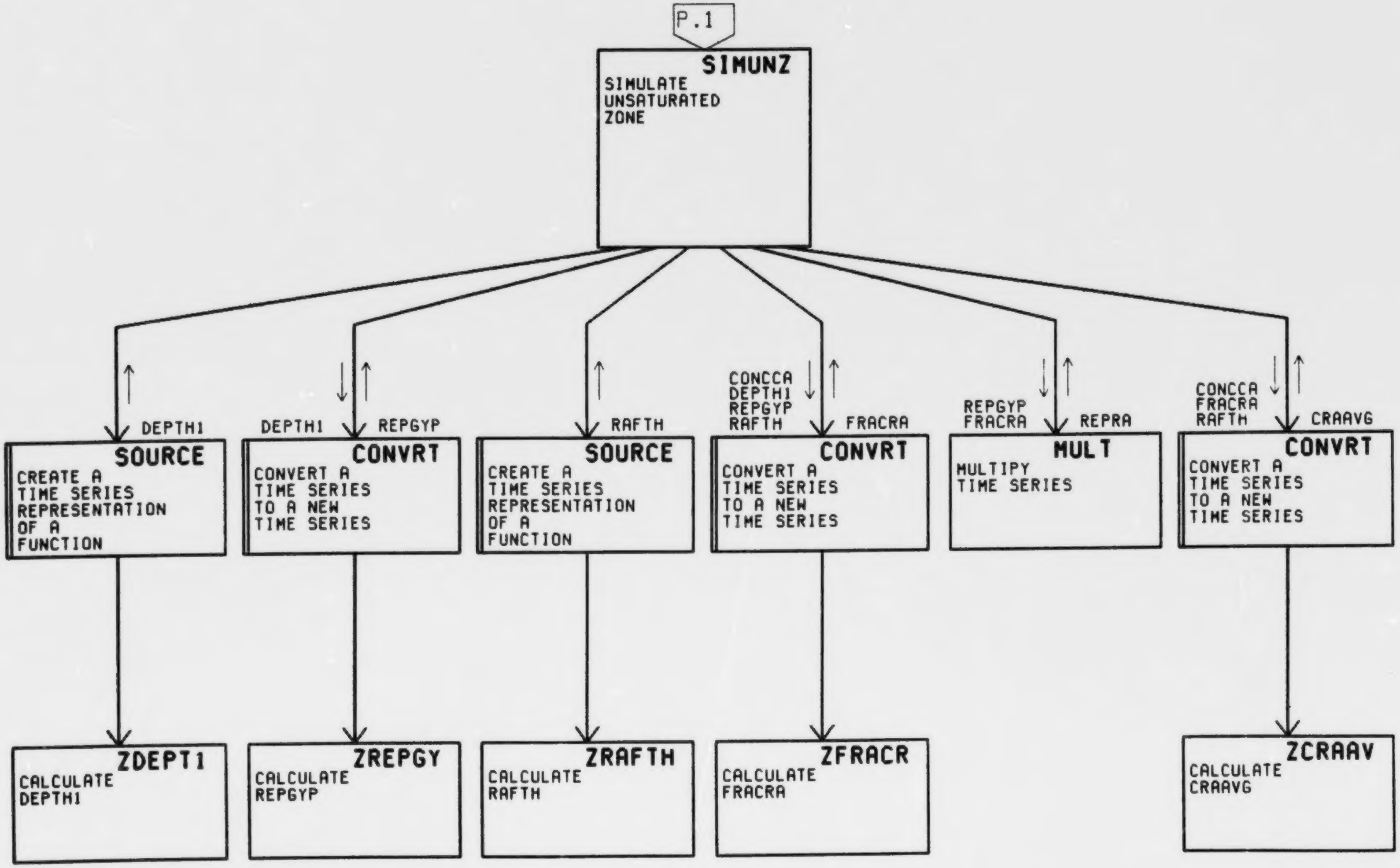

P.1
SIMUNZ
SIMULATE
UNSATURATED
ZONE
DEPTH1
SOURCE
CREATE A
TIME SERIES
REPRESENTATION
OF A
FUNCTION
DEPTH1
REPGYP
CONVRT
CONVERT A
TIME SERIES
TO A NEW
TIME SERIES
RAFTH
SOURCE
CREATE A
TIME SERIES
REPRESENTATION
OF A
FUNCTION
CONCCA
DEPTH1
REPGYP
RAFTH
FRACRA
CONVRT
CONVERT A
TIME SERIES
TO A NEW
TIME SERIES
REPGYP
FRACRA
REPRA
MULT
MULTIPY
TIME SERIES
CONCCA
FRACRA
RAFTH
CRAAVG
CONVRT
CONVERT A
TIME SERIES
TO A NEW
TIME SERIES
ZDEPT1
CALCULATE
DEPTH1
ZREPGY
CALCULATE
REPGYP
ZRAFTH
CALCULATE
RAFTH
ZFRACR
CALCULATE
FRACRA
ZCRAAV
CALCULATE
CRAAVG

FIGURE 7: Structure Chart for Subroutine SIMUNZ. The pentagonal box at the top of the figure shows that this structure chart is referred to by a higher level of structure charts; in this case it is structure chart 1, which is reproduced in Figure 5.

P.3

T. Andres 86-APR-02

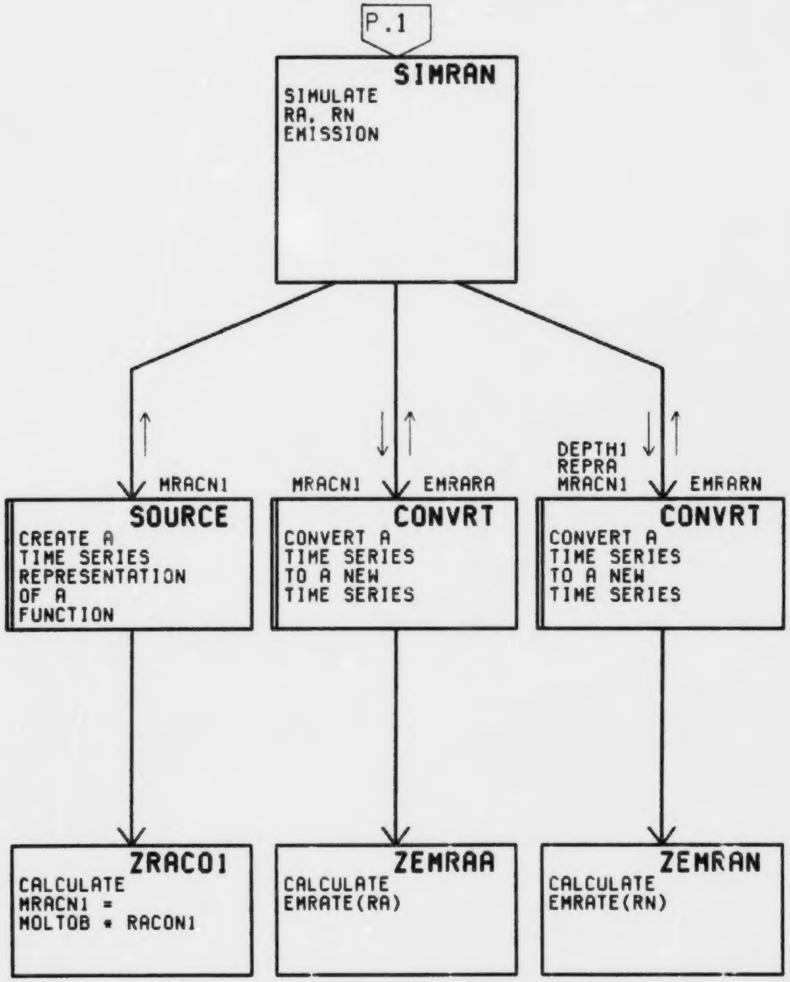

P.1
SIMRAN
SIMULATE
RA, RN
EMISSION
MRACN1
SOURCE
CREATE A
TIME SERIES
REPRESENTATION
OF A
FUNCTION
MRACN1
EMRARA
CONVRT
CONVERT A
TIME SERIES
TO A NEW
TIME SERIES
DEPTH1
REPRA
MRACN1
EMRARN
CONVRT
CONVERT A
TIME SERIES
TO A NEW
TIME SERIES
ZRACO1
CALCULATE
MRACN1 =
MOLTOB * RACON1
ZEMRAA
CALCULATE
EMRATE(RA)
ZEMRAN
CALCULATE
EMRATE(RN)

FIGURE 8: Structure Chart for Subroutine SIMRAN. The pentagonal box at the top of the figure shows that this structure chart is referred to be a higher level of structure charts; in this case it is structure chart 1, which is reproduced in Figure 5.

P.4

T. Andres 86-APR-02

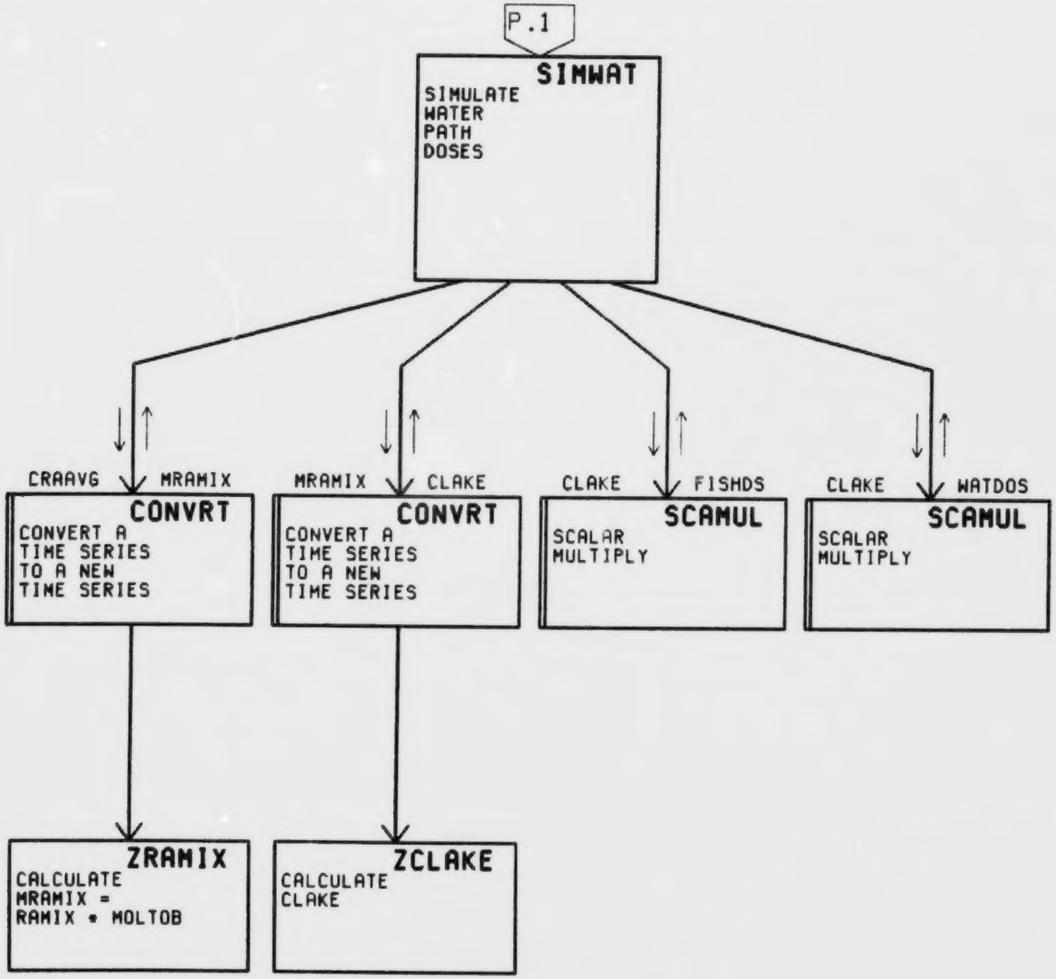

P.1
SIMWAT
SIMULATE
WATER
PATH
DOSES
CRAAVG
MRAMIX
CONVRT
CONVERT A
TIME SERIES
TO A NEW
TIME SERIES
MRAMIX
CLAKE
CONVRT
CONVERT A
TIME SERIES
TO A NEW
TIME SERIES
CLAKE
FISHDS
SCAMUL
SCALAR
MULTIPLY
CLAKE
WATDOS
SCAMUL
SCALAR
MULTIPLY
ZRAMIX
CALCULATE
MRAMIX =
RAMIX * MOLTOB
ZCLAKE
CALCULATE
CLAKE

FIGURE 9: Structure Chart for Subroutine SIMWAT. The pentagonal box at the top of the figure shows that this structure chart is referred to by a higher level of structure charts; in this case it is structure chart 1, which is reproduced in Figure 5.

P.5

T. Andres 86-APR-02

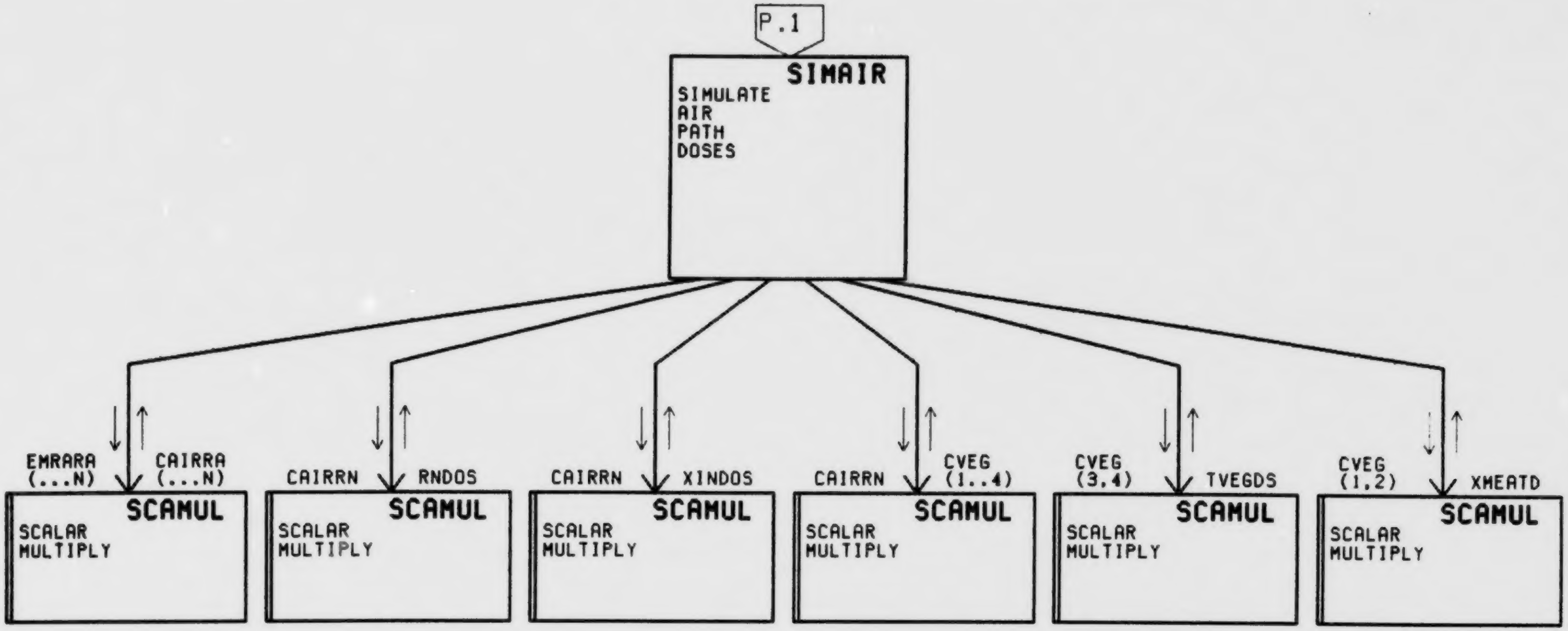

P.1
SIMAIR
SIMULATE
AIR
PATH
DOSES
EMRARA
(...N)
CAIRRA
(...N)
SCAMUL
SCALAR
MULTIPLY
CAIRRN
RNDOS
SCAMUL
SCALAR
MULTIPLY
CAIRRN
XINDOS
SCAMUL
SCALAR
MULTIPLY
CAIRRN
CVEG
(1..4)
SCAMUL
SCALAR
MULTIPLY
CVEG
(3,4)
TVEGDS
SCAMUL
SCALAR
MULTIPLY
CVEG
(1,2)
XMEATD
SCAMUL
SCALAR
MULTIPLY

FIGURE 10: Structure Chart for Subroutine SIMAIR. The pentagonal box at the top of the figure shows that this structure chart is referred to by a higher level of structure charts; in this case it is structure chart 1, which is reproduced in Figure 5.

P.6

T. Andres 86-APR-02

APL was chosen because it is an interactive language with powerful operators. "Number-crunching" programs written in APL are typically one tenth the size of corresponding FORTRAN programs (and one tenth the speed). Programs can be implemented and tested quickly in APL because they are short and because they can be stopped and examined in detail at critical points within the computations. It was felt, therefore, that a second implementation in APL would be an efficient use of resources in testing the FORTRAN code. The APL implementation would not itself be useful for a code comparison, however, because it was slow, incomplete, and difficult to follow and maintain by inexperienced users.

The APL version of the model operated as planned. It was implemented in about two days, and was valuable in calibrating and correcting the FORTRAN version. The procedure followed was to compare the results of the two codes on one random set of parameter values, and to adjust the codes until they agreed. Then the same procedure was followed on a second random set of parameter values. Finally, on the third set, they agreed without adjustment.

A preliminary decision made with regard to the APL implementation later proved to be a mistake. It was originally decided that the APL model would not solve complicated equations, but would simply test that values produced by SYVAC3 were appropriate solutions to the equations. Later, when detailed comparison with UTAP results began, some errors in the SYVAC3 results were discovered that had not been detected by this procedure. It was therefore decided that the APL implementation was to perform all calculations and solve all equations exactly, for comparison with the SYVAC3 implementation.

Finding solutions to complicated equations was the most technically difficult aspect of the code development. It was decided early that a very reliable technique, rather than a fast one, would be used to solve equations such as the one for TFINAL in specification 1.2.2 or the one for PHLAKE in specification 4.3.1. The method chosen was the bisection technique described in Hamming (1971). If a root of an equation has been localized to an interval, then the bisection technique approximates the root to any desired accuracy in a precisely determined number of function evaluations. This procedure worked very well, except when the root could not be localized to an interval. This difficulty occurred with PHLAKE and with FRACRA, both at time 0. In the case of FRACRA, the equation being evaluated had an indeterminate form (i.e., 0/0) at time 0. The problems with both equations were later traced to incorrect specifications.

When the APL model was modified to solve these equations, an equally reliable, but faster, algorithm was found and used. It was the Van Wijngaarden-Dekker-Brent method from Press et al. (1986). This technique required fewer function evaluations than pure bisection, and could have been implemented in the FORTRAN code as well if execution time had been a problem.

During the implementation of the uranium tailings model, a few poorly characterized errors were detected, which proved to be difficult to find. These tended to be errors in the time series routines (and in one case a compiler error). They had not been found earlier because quality assurance of the SYVAC3 executive had not yet included the time series routines.

In summary, the final result of the coding and testing phase was the development of two implementations of the uranium tailings model. The APL and FORTRAN implementations were both developed from the same specifications (reproduced in Appendix B), and both were cross-checked until each produced equivalent results.

3.0 COMPARISON OF UTAP AND SYVAC3

3.1 Comparison Criteria

Prior to the start of the analysis of UTAP and SYVAC3 results, a set of comparison criteria were proposed. These criteria were to be used to provide a summary of how well the results agreed when the two codes were applied to an assessment of a generic uranium tailings site.

For the comparison of simple computer models, it could be argued that all results should agree to a large number of significant figures. This was not the case here, however, since the uranium tailings model is relatively complicated, and it would be embedded within the long and complicated executive modules of UTAP and SYVAC3. As a consequence, exact agreement of the results was not expected, nor was it essential. For this comparison, it would be adequate to demand that the two codes reproduce the same trends, and that the results be "reasonably" similar. More precise definitions of the proposed comparison criteria are given below.

For both the deterministic and probabilistic runs, the comparison criteria were based on the major output variables from the uranium tailings model:

(1) pH of the lake (PHLAKE)
(2) Total annual dose (TDOSE)
(3) Dose from ingestion of vegetables (TVEGDS)
(4) Dose from ingestion of caribou (XMEATD)
(5) Dose from ingestion of fish (FISHDS)
(6) Dose from ingestion of water (WATDOS)
(7) Dose from inhalation of dust (XINDOS)
(8) Dose from inhalation of radon (RNDOS)

Values of these eight variables were stored for comparison at 11 different times, in 50-year increments from 0 to 500 years. A total of 88 variables were therefore available for comparison.

For the single deterministic run, the following set of comparison criteria were defined in terms of number of significant figures of agreement:

"good agreement" -
All 88 variables agree to within two significant figures or more, when rounded off.

"fair agreement" -
At least 85 of the 88 variables agree to within two significant figures or more, when rounded off. One to three variables may agree to within only one significant figure, when rounded off.

All plots show the same trends, although the same one to three variables could have local extrema and/or asymptotic limits that agree to within only one significant figure, when rounded off.

"poor agreement" -
More than 3 of the 88 variables do not agree to within two significant figures or more, when rounded off, and/or the plots for more than three variables do not show the same trends.

As an operational definition, one significant figure of agreement is interpreted as 20% error or less; two significant figures as 2% error or less.

For the probabilistic runs, the proposed comparisons were to involve the mean values of the 88 variables. Tests could also be based on observed standard deviations, but at the time these criteria were proposed, only four standard deviations were available from the SENES results. SENES subsequently produced a table of statistics for their 500 simulations (reproduced in Appendix A). Accordingly, the criteria for probabilistic runs have been changed to accommodate these extra statistics. The probabilistic comparisons were based on the t-test to compare geometric means and the F-test to compare geometric standard deviations. Two statistics, one from UTAP and one from SYVAC3, were said to disagree if the hypothesis of agreement could be rejected at the 5% significance level[2] . Otherwise they were said to agree. With this modification to the original criteria, the qualitative comparison summaries were defined as follows:

"good agreement" -
The mean values and standard deviations of at least 80 of the 88 variables agree as defined above.

"fair agreement" -
The mean values and standard deviations of at least 76 of the 88 variables agree as defined above.

"poor agreement" -
The mean values and standard deviations of fewer than 76 of the 88 variables agree as defined above.

3.2 Comparison of Deterministic Results

When the SYVAC3 code with the uranium tailings system model had been thoroughly tested, a single deterministic simulation was set up, using the same input data as used by SENES in their deterministic run (reproduced in Appendix A). The SYVAC3 deterministic results were then compared with the corresponding deterministic results provided by SENES (see Appendix A). The initial results are shown in Table 1. The agreement was poor by the criteria previously stated, because none of 88 variables matched SENES results to two significant figures.

Through consultation with SENES, changes were made to the SYVAC3 model and gradually the agreement improved. (A full record of the

---

2/ The significance level is the probability of falsely rejecting a null hypothesis in a statistical test.

TABLE 1. First Comparison of SYVAC3 and UTAP Results for the Deterministic Run[1]

	Time [a]										
	0	50	100	150	200	250	300	350	400	450	500
FISH INGESTION DOSE											
SYVAC	0.00E000	3.08E-07	1.35E-06	9.52E-06	9.44E-06	8.04E-06	6.81E-06	5.77E-06	4.90E-06	4.17E-06	3.56E-06
UTAP	1.75E000	2.40E000	3.69E000	1.15E001	1.03E001	8.72E000	7.37E000	6.24E000	5.29E000	4.50E000	3.84E000
DIFF.	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)
CARIBOU MEAT INGESTION DOSE											
SYVAC	1.42E-06	1.40E-06	1.38E-06	1.36E-06	1.34E-06	1.32E-06	1.30E-06	1.29E-06	1.27E-06	1.25E-06	1.23E-06
UTAP	1.45E-03	1.43E-03	1.41E-03	1.39E-03	1.37E-03	1.35E-03	1.33E-03	1.31E-03	1.30E-03	1.28E-03	1.26E-03
DIFF.	(99.90%)	(99.90%)	(99.90%)	(99.90%)	(99.90%)	(99.90%)	(99.90%)	(99.90%)	(99.90%)	(99.90%)	(99.90%)
RADON INHALATION DOSE											
SYVAC	7.71E-06	6.81E-06	6.43E-06	5.91E-06	5.83E-06	5.75E-06	5.67E-06	5.59E-06	5.52E-06	5.44E-06	5.37E-06
UTAP	7.71E000	6.81E000	6.42E000	5.91E000	5.83E000	5.75E000	5.67E000	5.59E000	5.52E000	5.44E000	5.37E000
DIFF.	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)
VEGETATION INGESTION DOSE											
SYVAC	2.43E-06	2.40E-06	2.36E-06	2.33E-06	2.30E-06	2.27E-06	2.23E-06	2.20E-06	2.17E-06	2.15E-06	2.12E-06
UTAP	2.43E000	2.40E000	2.36E000	2.33E000	2.30E000	2.27E000	2.23E000	2.21E000	2.17E000	2.15E000	2.12E000
DIFF	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)
WATER INGESTION DOSE											
SYVAC	0.00E000	8.67E-07	3.81E-06	2.68E-05	2.66E-05	2.26E-05	1.91E-05	1.62E-05	1.38E-05	1.17E-05	1.00E-05
UTAP	4.93E000	6.75E000	1.04E001	3.24E001	2.90E001	2.45E001	2.07E001	1.76E001	1.49E001	1.27E001	1.08E001
DIFF	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)
DUST INHALATION DOSE											
SYVAC	1.08E-06	1.07E-06	1.05E-06	1.04E-06	1.02E-06	1.01E-06	9.94E-07	9.80E-07	9.67E-07	9.54E-07	9.42E-07
UTAP	1.08E000	1.07E000	1.05E000	1.04E000	1.02E000	1.01E000	9.94E-01	9.81E-01	9.68E-01	9.55E-01	9.42E-01
DIFF.	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)
TOTAL DOSE											
SYVAC	1.26E-05	1.28E-05	1.64E-05	4.69E-05	4.65E-05	4.10E-05	3.61E-05	3.20E-05	2.86E-05	2.57E-05	2.32E-05
UTAP	1.79E001	1.94E001	2.39E001	5.31E001	4.84E001	4.23E001	3.70E001	3.26E001	2.89E001	2.57E001	2.31E001
DIFF.	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)	(100.00%)
LAKE pH											
SYVAC	3.88E000	3.87E000	3.94E000	4.03E000	4.08E000	4.10E000	4.12E000	4.15E000	4.18E000	4.21E000	4.24E000
UTAP	4.00E000	4.15E000	4.42E000	6.75E000	7.15E000	7.23E000	7.26E000	7.26E000	7.27E000	7.27E000	7.27E000
DIFF	(3.02%)	(6.83%)	(10.84%)	(40.32%)	(42.96%)	(43.22%)	(43.15%)	(42.90%)	(42.55%)	(42.14%)	(41.65%)

---

[1] DIFF is the percentage difference between SYVAC and UTAP results, calculated using

$$\frac{(\text{UTAP}-\text{SYVAC})}{\text{UTAP}} \times 100\% .$$

consultations are provided in Appendix A.) A summary of the major changes required, and the resultant number of variables agreeing to within 20% or 2% are as follows:

(1) First deterministic comparison.
Results: 3 variables had differences of less than 20%; none had differences of less than 2%.

(2) Corrected units for dose.
Results: 60 variables differed by less than 20%; 37 by less than 2%.

(3) Corrected errors in pH calculations (Minispecification figures 4.3.1, 4.3.2, and 4.3.3):
   - one equation error in program;
   - two equation errors in SENES documents;
   - two constants with wrong units in specifications;
   - one incorrect constant value in specifications.

   First attempt to match the assumption stated by SENES that processes begin in the time step before t = 0, rather than at t = 0.
   Results: 62 variables differed by less than 20%; 37 by less than 2%.

(4) Corrected initial conditions to match SENES undocumented initial conditions for SULAKE, CALAKE, RAMIX, MLAKE, and MSED. Corrected value of constant parameter FIR. Corrected error in specifications (1.2.3.5) due to ambiguous documentation.
Results: 72 variables differed by less than 20%; 46 by less than 2%.

(5) Recovered from complete failure of hard disk, and loss of programs and data. Backups had to be re-installed and tested. Corrected initial condition to match SENES initial condition for FRACRA (1.4.1). Corrected two errors in input file introduced while recovering from disk failure.
Final results: All 88 variables differed by less than 20%; 48 by less than 2%.

No further improvements were made because of shortage of time. The final outputs of the SYVAC3 code were qualitatively similar to UTAP outputs, in that the largest percentage difference was less than 20%, but the agreement was still "poor" according to the original criteria. The final results are compared in detail in Table 2.

In spite of this "poor" agreement, some overall results of the comparison are worth noting. The doses due to inhalation of radon and dust, the doses due to ingestion of vegetables, and the lake pH values are almost identical; these 44 variables show less than 0.5% difference between UTAP and SYVAC3 results. The 11 caribou meat ingestion doses agreed to within about 2.2%. The remaining 33 values showing the largest differences are those associated with the water pathways, i.e., doses due to ingestion of fish and water and, hence, the total dose. The percentage difference for these variables was less than 5%, except for the 100 year values.

TABLE 2. Final Comparison of SYVAC3 and UTAP Results for the Deterministic Run[1]

Time [a]

	0	50	100	150	200	250	300	350	400	450	500
FISH INGESTION DOSE											
SYVAC	1.75E000	2.30E000	3.04E000	1.09E001	1.06E001	9.04E000	7.64E000	6.46E000	5.48E000	4.66E000	3.97E000
UTAP	1.75E000	2.40E000	3.69E000	1.15E001	1.03E001	8.72E000	7.37E000	6.24E000	5.29E000	4.50E000	3.84E000
DIFF	(0.03%)	(4.19%)	(17.56%)	(4.99%)	(3.12%)	(3.60%)	(3.59%)	(3.56%)	(3.51%)	(3.46%)	(3.40%)
CARIBOU MEAT INGESTION DOSE											
SYVAC	1.42E-03	1.40E-03	1.38E-03	1.36E-03	1.34E-03	1.32E-03	1.30E-03	1.29E-03	1.27E-03	1.25E-03	1.23E-03
UTAP	1.45E-03	1.43E-03	1.41E-03	1.39E-03	1.37E-03	1.35E-03	1.33E-03	1.31E-03	1.30E-03	1.28E-03	1.26E-03
DIFF	(2.18%)	(2.22%)	(2.16%)	(2.20%)	(2.23%)	(2.22%)	(2.18%)	(2.18%)	(2.16%)	(2.19%)	(2.19%)
RADON INHALATION DOSE											
SYVAC	7.71E000	6.81E000	6.42E000	5.91E000	5.83E000	5.75E000	5.67E000	5.59E000	5.52E000	5.44E000	5.37E000
UTAP	7.71E000	6.81E000	6.42E000	5.91E000	5.83E000	5.75E000	5.67E000	5.59E000	5.52E000	5.44E000	5.37E000
DIFF	(0.04%)	(0.02%)	(0.02%)	(0.04%)	(0.03%)	(0.03%)	(0.03%)	(0.05%)	(0.03%)	(0.03%)	(0.03%)
VEGETATION INGESTION DOSE											
SYVAC	2.43E000	2.40E000	2.36E000	2.33E000	2.30E000	2.27E000	2.23E000	2.20E000	2.17E000	2.15E000	2.12E000
UTAP	2.43E000	2.40E000	2.36E000	2.33E000	2.30E000	2.27E000	2.23E000	2.21E000	2.17E000	2.15E000	2.12E000
DIFF	(0.05%)	(0.03%)	(0.05%)	(0.02%)	(0.03%)	(0.02%)	(0.02%)	(0.03%)	(0.02%)	(0.02%)	(0.05%)
WATER INGESTION DOSE											
SYVAC	4.93E000	6.47E000	8.56E000	3.08E001	2.99E001	2.54E001	2.15E001	1.82E001	1.54E001	1.31E001	1.12E001
UTAP	4.93E000	6.75E000	1.04E001	3.24E001	2.90E001	2.45E001	2.07E001	1.76E001	1.49E001	1.27E001	1.08E001
DIFF	(0.04%)	(4.20%)	(17.59%)	(4.99%)	(3.18%)	(3.58%)	(3.59%)	(3.53%)	(3.53%)	(3.47%)	(3.44%)
DUST INHALATION DOSE											
SYVAC	1.08E000	1.07E000	1.05E000	1.04E000	1.02E000	1.01E000	9.94E-01	9.80E-01	9.67E-01	9.54E-01	9.42E-01
UTAP	1.08E000	1.07E000	1.05E000	1.04E000	1.02E000	1.01E000	9.94E-01	9.81E-01	9.68E-01	9.55E-01	9.42E-01
DIFF	(0.07%)	(0.01%)	(0.02%)	(0.09%)	(0.03%)	(0.03%)	(0.04%)	(0.04%)	(0.04%)	(0.03%)	(0.04%)
TOTAL DOSE											
SYVAC	1.79E001	1.90E001	2.14E001	5.10E001	4.97E001	4.35E001	3.80E001	3.34E001	2.96E001	2.63E001	2.36E001
UTAP	1.79E001	1.94E001	2.39E001	5.31E001	4.84E001	4.23E001	3.70E001	3.26E001	2.89E001	2.57E001	2.31E001
DIFF	(0.04%)	(1.97%)	(10.34%)	(4.11%)	(2.56%)	(2.81%)	(2.73%)	(2.59%)	(2.44%)	(2.30%)	(2.13%)
LAKE pH											
SYVAC	4.00E000	4.15E000	4.42E000	6.77E000	7.16E000	7.24E000	7.26E000	7.27E000	7.27E000	7.27E000	7.27E000
UTAP	4.00E000	4.15E000	4.42E000	6.75E000	7.15E000	7.23E000	7.26E000	7.26E000	7.27E000	7.27E000	7.27E000
DIFF	(0.02%)	(0.04%)	(0.05%)	(0.36%)	(0.18%)	(0.15%)	(0.05%)	(0.02%)	(0.00%)	(0.00%)	(0.00%)

---

[1] In terms of the operational comparison criteria, 48 variables agreed to within 2% or less, and all 88 to within 20% or less. DIFF is the percentage between SYVAC and UTAP results, calculated using

$$\frac{(\text{UTAP}-\text{SYVAC})}{\text{UTAP}} \times 100\%.$$

3.3 Comparison of Probabilistic Results

After the deterministic comparison was completed, UTAP and SYVAC3 results were compared for a set of 500 independent random simulations. Unlike the deterministic comparison, this comparison did not require both codes to run exactly the same simulations. Instead, they ran the same number of simulations using the same input probability distributions (but not the same sets of randomly selected values), and the statistical properties of the results were compared.

It might be expected that the results would necessarily compare poorly, given that the deterministic comparison had already given poor results. This inference is not necessarily justified, however, since the deterministic outputs of the two codes are qualitatively similar. Conceivably slight differences between the codes could be swamped by statistical variability. Furthermore, the criteria in the two comparisons differed. The probabilistic criteria were designed to allow more scope for variation because statistical behaviour was involved. Good agreement between the two codes was therefore possible, and so the statistical test was important in its own right.

Nevertheless, it was not surprising to find that the comparison of statistical results showed "poor" agreement (as defined in Section 3.1), between UTAP and SYVAC3 outputs, as shown in Table 3. Specifically, only 51 of the 88 consequence variables showed agreement in both the mean and standard deviation. The remaining 37 consequences were statistically different (at the 5% significance level) in either the mean or standard deviation or both. Of these, 23 differed in means only, 1 differed in standard deviations only, and 13 differed in both means and standard deviations. Plots of the mean dose variables as a function of time, for both the SYVAC3 and UTAP results, are shown in Figure 11. Both codes predict the same trends in all cases.

These comparison results differed from the deterministic comparison results primarily for two variables. First, doses from ingestion of caribou meat agreed in the probabilistic comparison, even though they disagreed by just over 2% in the deterministic comparison. Second, lake pH values past 150 years differed statistically in mean and standard deviation in the probabilistic comparison, even though they matched very well in the deterministic comparison. (The pH differences may, in fact, be due to round-off and/or truncation, since the standard deviations were all very small, and only four significant figures were available from the UTAP results.)

The trends in the UTAP-SYVAC3 comparison found in the deterministic runs were also evident in the probabilistic runs. Generally speaking, all dose values agreed quite well, except for those associated with the water pathway (ingestion of fish and water and, hence, total dose) and the mean and standard deviation in lake pH values (beyond 150 years).

3.4 Summary of the Comparisons

On the basis of the original acceptance criteria, the agreement between UTAP results and SYVAC results would be characterized as "poor". However, the results from the two codes were qualitatively similar for all

TABLE 3. Comparison of SYVAC3 and UTAP Statistics for the Probabilistic Run[1] (Only SYVAC results are shown).

						Time [a]					
	0	50	100	150	200	250	300	350	400	450	500
FISH INGESTION DOSE											
GM	2.73E000	2.47E000	3.38E000	7.61E000	1.12E001	9.08E000	6.94E000	5.44E000	4.41E000	3.67E000	3.12E000
GSD	3.56E000	3.61E000	3.92E000	4.90E000	4.56E000	4.40E000	4.38E000	4.47E000	4.57E000	4.67E000	4.75E000
P_1	(0.000)	(0.716)	(0.008)	(0.052)	(0.047)	(0.004)	(0.003)	(0.003)	(0.004)	(0.005)	(0.007)
P_2	(0.484)	(0.492)	(0.152)	(0.453)	(0.481)	(0.708)	(0.923)	(0.765)	(0.741)	(0.772)	(0.823)
CARIBOU MEAT INGESTION DOSE											
GM	8.15E-04	8.04E-04	7.93E-04	7.83E-04	7.73E-04	7.63E-04	7.54E-04	7.44E-04	7.35E-04	7.26E-04	7.17E-04
GSD	4.28E000	4.28E000	4.28E000	4.28E000	4.28E000	4.28E000	4.28E000	4.28E000	4.28E000	4.28E000	4.28E000
P_1	(0.238)	(0.238)	(0.238)	(0.239)	(0.239)	(0.239)	(0.240)	(0.240)	(0.240)	(0.241)	(0.241)
P_2	(0.303)	(0.303)	(0.301)	(0.301)	(0.299)	(0.297)	(0.296)	(0.296)	(0.294)	(0.294)	(0.294)
RADON INHALATION DOSE											
GM	7.55E000	6.39E000	5.96E000	5.39E000	5.12E000	5.05E000	4.98E000	4.92E000	4.86E000	4.80E000	4.74E000
GSD	2.44E000	2.55E000	2.55E000	2.49E000	2.45E000	2.45E000	2.45E000	2.45E000	2.44E000	2.44E000	2.44E000
P_1	(0.924)	(0.970)	(0.977)	(0.908)	(0.893)	(0.891)	(0.890)	(0.891)	(0.893)	(0.894)	(0.896)
P_2	(0.584)	(0.660)	(0.725)	(0.750)	(0.886)	(0.878)	(0.878)	(0.869)	(0.877)	(0.869)	(0.869)
VEGETATION INGESTION DOSE											
GM	1.97E000	1.95E000	1.92E000	1.90E000	1.87E000	1.85E000	1.83E000	1.80E000	1.78E000	1.76E000	1.74E000
GSD	2.27E000	2.27E000	2.27E000	2.27E000	2.27E000	2.27E000	2.27E000	2.27E000	2.27E000	2.26E000	2.26E000
P_1	(0.897)	(0.904)	(0.895)	(0.901)	(0.900)	(0.899)	(0.906)	(0.905)	(0.912)	(0.911)	(0.910)
P_2	(0.365)	(0.364)	(0.364)	(0.370)	(0.370)	(0.369)	(0.362)	(0.362)	(0.361)	(0.361)	(0.360)
WATER INGESTION DOSE											
GM	7.59E000	6.85E000	9.38E000	2.11E001	3.12E001	2.52E001	1.93E001	1.51E001	1.23E001	1.02E001	8.67E000
GSD	1.55E000	1.60E000	1.96E000	2.79E000	2.67E000	2.44E000	2.40E000	2.46E000	2.55E000	2.62E000	2.67E000
P_1	(0.000)	(0.103)	(0.000)	(0.001)	(0.004)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
P_2	(0.445)	(0.039)	(0.000)	(0.203)	(0.003)	(0.056)	(0.440)	(0.660)	(0.618)	(0.517)	(0.427)
DUST INHALATION DOSE											
GM	9.42E-01	9.30E-01	9.18E-01	9.06E-01	8.94E-01	8.83E-01	8.72E-01	8.61E-01	8.50E-01	8.39E-01	8.29E-01
GSD	1.39E000	1.39E000	1.39E000	1.38E000	1.38E000	1.38E000	1.38E000	1.38E000	1.38E000	1.38E000	1.38E000
P_1	(1.000)	(0.996)	(0.992)	(0.992)	(0.987)	(0.982)	(0.982)	(0.977)	(0.973)	(0.967)	(0.967)
P_2	(0.697)	(0.696)	(0.696)	(0.695)	(0.694)	(0.729)	(0.729)	(0.728)	(0.727)	(0.726)	(0.726)
TOTAL DOSE											
GM	2.55E001	2.31E001	2.75E001	4.83E001	6.35E001	5.31E001	4.33E001	3.65E001	3.17E001	2.81E001	2.55E001
GSD	1.64E000	1.66E000	1.80E000	2.42E000	2.42E000	2.23E000	2.10E000	2.05E000	2.02E000	2.01E000	2.00E000
P_1	(0.000)	(0.368)	(0.000)	(0.007)	(0.006)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.001)
P_2	(0.869)	(0.812)	(0.001)	(0.227)	(0.000)	(0.001)	(0.015)	(0.058)	(0.094)	(0.104)	(0.107)
LAKE pH											
MEAN	4.00E000	4.16E000	4.56E000	6.41E000	7.12E000	7.22E000	7.25E000	7.26E000	7.27E000	7.27E000	7.27E000
SD	2.55E-02	1.15E-02	4.31E-01	8.67E-01	1.98E-01	6.02E-02	2.56E-02	1.26E-02	6.66E-03	3.69E-03	2.13E-03
P_1	(0.551)	(0.170)	(0.857)	(0.133)	(0.000)	(0.000)	(0.000)	(0.001)	(0.041)	(0.000)	(0.000)
P_2	(0.119)	(0.846)	(0.448)	(0.433)	(0.000)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

---

[1] P_1 = significance level of t-statistic GSD = geometric standard deviation

P_2 = significance level of F-statistic GM = geometric mean

There is evidence of significant differences between SYVAC and UTAP if $P_1 < 0.05$ or $P_2 < 0.05$.

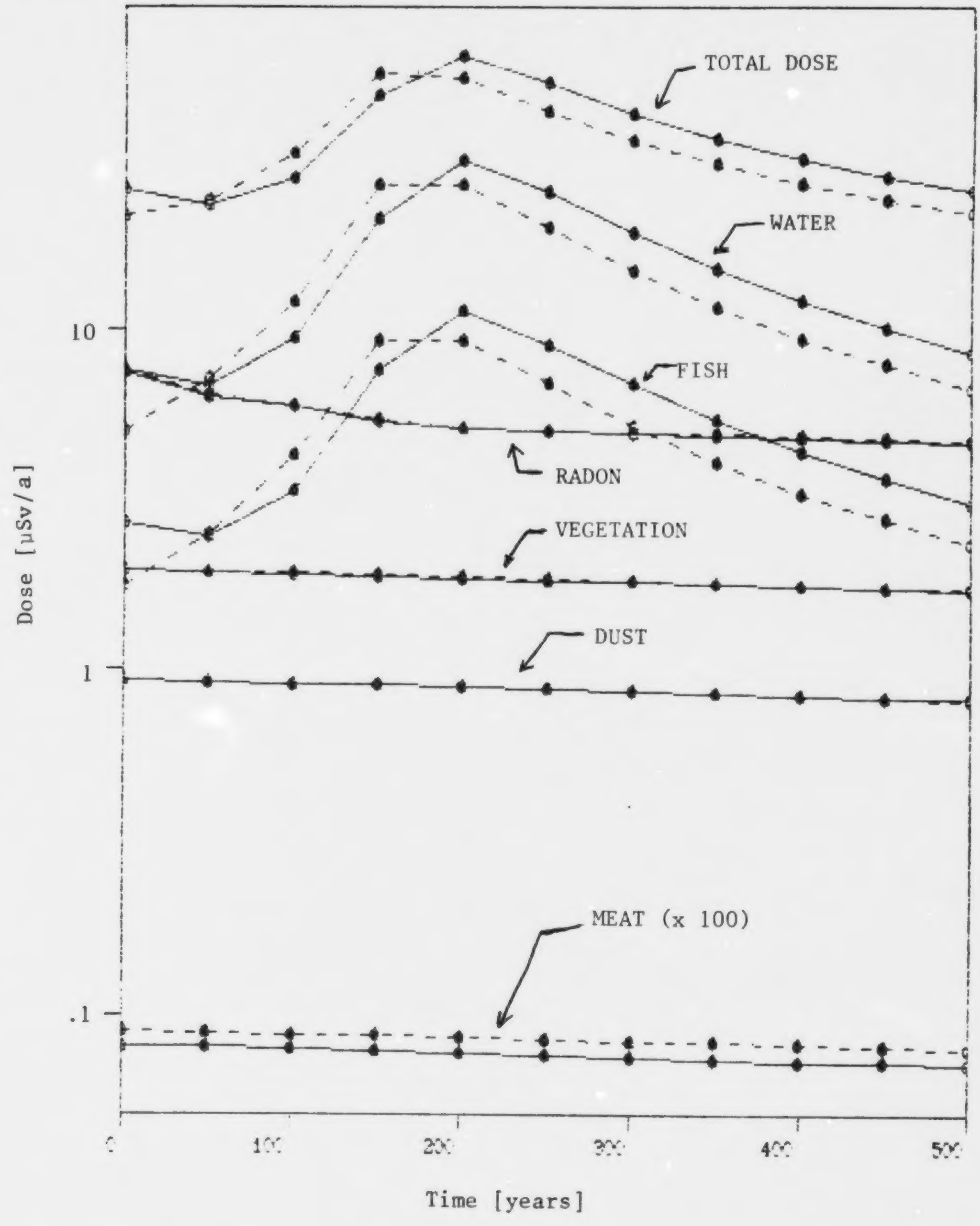


FIGURE 11: Plots of the Dose Variables as a Function of Time. The seven dose variables are geometric means for total dose, dose due to the ingestion of contaminated drinking water, fish, caribou meat and vegetation, and dose due to the inhalation of radium in dust and radon gas and its daughters. The solid curves are from SYVAC3 results, and the dashed curves from UTAP results (SENES, 1985).

variables, and quantitatively very close for many of the variables. This confusing result means that there were minor differences between the two codes that could not be readily identified on the basis of the data available.

The remaining differences between UTAP and SYVAC3 occurred primarily in the water pathways, i.e. doses from ingestion of fish and water. In addition, there were small differences in the terrestrial pathway for doses from ingestion of caribou meat (deterministic comparison only). Since all doses contribute to total dose, the total doses also differed at most times in both comparisons.

The remaining differences are also probably due to a number of small discrepancies, rather than to one or two large ones. Evidence for this statement is found in the dispersed nature of the disagreements (i.e., disagreements were found for several variables, and depended on the type of comparison). This conclusion is also justified by the pattern of differences discovered in the deterministic comparison. As described in Section 3.2, attempts to make UTAP and SYVAC3 results agree uncovered multiple discrepancies, including incorrect units, errors in constants and unspecified initial conditions. In some cases, a number of these discrepancies were eliminated with little effect on the agreement of results. In other cases, however, a few discrepancies were responsible for most of the disagreement, but a larger number of discrepancies had only minor effects. Since only relatively minor differences remain, we conclude that several relatively small discrepancies still exist.

With more effort, all the remaining differences between the two codes could be eliminated, although it was not possible to do so within the time-frame of this contract. We recommend the following plan of action to eliminate all remaining differences:

(1) Run UTAP and prepare six deterministic simulations, in each one printing out all intermediate variables, such as concentrations of each nuclide at intermediate locations along each pathway at times 0, 50, 100, ... 500 years.

(2) Run the same deterministic simulations using SYVAC3.

(3) Compare all values of the variables produced by UTAP and SYVAC3 for one of the deterministic simulations. Investigate differences, first by checking that the coding of the uranium tailings model in SYVAC3 matches its specifications and then that the specifications match UTAP documentation. Finally, if necessary, discuss details of the suspect part of the uranium tailings model with SENES.

(4) Track down and eliminate differences until UTAP and SYVAC3 results agree for a given simulation.

(5) When, finally, results from the two codes agree on the deterministic simulation, compare the next deterministic simulation. If differences are found, repeat steps 3 and 4. If no differences are found, compare any remaining deterministic simulations

to ensure that the two codes are compared over a variety of values of the input variables.

(6) Run the probabilistic simulations. The probabilistic comparison may well yield agreement. If not, differences in the results may be due either to differences in models or to differences in the executive portions of the codes (for example, in the algorithms used to generate sample random values). Try deterministic comparisons of extreme-valued runs from the two random samples first, to identify possible differences in models. If no differences are found there, then check the behaviour of the executive modules.

This procedure (up to step 5) is the one we used to test the FORTRAN implementation against the APL implementation, and it is essentially what we used to test SYVAC3 against UTAP. Two limitations made the SYVAC3-UTAP comparison especially difficult: (1) only one deterministic run was available and (2) few of the intermediate variables were documented in UTAP results. In planning any future code comparisons, these limitations should be avoided from the start.

The comparison could also be continued by comparing listings of the two codes directly. This approach would provide a useful supplement to the comparison procedure described above. However, as contractors performing such a comparison, we feel that the necessary independence can best be maintained by comparing only results, and not the codes themselves. Direct comparison of the codes, for example, could lead to the copying of significant assumptions, from one code to the other, that are not reproduced in the specifications. This last problem could also occur in the procedure described above (to eliminate all remaining differences), and must be guarded against, to ensure quality assurance requirements are satisfied.

4.0 SUMMARY AND CONCLUSIONS

The original intention of this contract was to compare the UTAP (Uranium Tailings Assessment Program) code developed by SENES Consultants with a version of the SYVAC3 code linked to the same model of a uranium tailings system. The SYVAC3 code was to be quality-assured to ensure that the comparison was valid. The final results of this work would be to provide additional verification for the UTAP software.

The uranium tailings code developed for use with SYVAC3 was based upon Appendices A to F of Volume 3 of the SENES Phase II report (SENES, 1985). It was assumed that this description of the UTAP code was complete and accurate. This assumption proved to be incorrect for the purposes of using the UTAP documentation to prepare an equivalent uranium tailings model for SYVAC3. During the course of the project, it became necessary to obtain supplemental information so that the uranium tailings model used in SYVAC3 would more closely match that used in UTAP.

Table 4 provides examples of some of the deficiencies that were found in the UTAP documentation (a more complete list of problems encountered is given in Appendices A to C inclusive). With the assistance of

TABLE 4. Summary of Major Types of Deficiencies in the Documentation of the Uranium Tailings Model (see also Appendices A to C of this report)

Type of Deficiency	Example*
Erroneous data values	- Table 5.1 (Volume 3, SENES (1985)) is not consistent with the data in Appendices A to F - Data for the feed ingestion rate (FIR) in Table E.2 of Appendix E is different from the value used in the probabilistic run. - Data for other variables, such as FRACGP, have been changed from the values given in Appendices A to F.
Units	- Units given for RACON1 and RACON2 are inconsistent where these variables are used in Appendices A and B. Similarily, units for RAMIX are inconsistent in Appendices A and C. - Units are missing for constants used in some equations, such as the factor 1.5 in Eq. (1.1) of Appendix B.
Erroneous equations	- Problems were detected in a number of equations; for example, in Appendix A the following equations were found to contain errors: Eqs. (3.17), (3.20), (3.21), (3.23), (7.9) and (7.10). - Several equations had more serious errors: for example, Eq. (7.11) in Appendix A, and the equation following Eq. (3.2) in Appendix C were incorrect. - SENES noted corrections, not detected in this report, to pH equations such as Eq. (7.8) in Appendix A and the equation on page C-12 of Appendix C. These equations could not be readily verified, given the available documentation.
Equations missing	- No equation is given for $[Ra]_{\lambda_w}$ = CRAAVG, referred to on page A-29 of Appendix A, for time less than TFINAL. Equation (A-19) in Appendix A covers time greater than or equal to TFINAL. - No equation is given for XK2, referenced in Appendix C.
Initial conditions missing or erroneous	- Initial conditions are missing or erroneous for most of the finite difference equations; for example, the initial conditions for Eqs. (2.5) and (2.6) in Appendix C were later corrected by SENES, and no initial conditions were provided for FRACRA in Eq. (3.24) of

...continued

TABLE 4 (Concluded)

Type of Deficiency	Example*
	Appendix A. SENES supplied several different initial conditions for FRACRA (see Appendix A of this report).
Notation varies	- A number of variables are described using different notations throughout the documentation, with no cross-referencing. CONCS8, for example, is referred to as C_s, $C_s(t,\lambda_w)$, $C(t,z)$, $[SO_4^{2-}]_p$ and $[SO_4^{2-}]_{\lambda w}$ at different locations in Appendix A.
Inconsistent definitions	- RAMIX in Appendix A is denoted as $[Ra]_{mix}$ and defined to be the "porewater concentration of radium in the mixing zone". In Appendix C, the notation is changed (presumably) to $[Ra]_{T,t}$ and the new definition is "concentration of radium in the contaminated surface runoff from the tailings area". The units of RAMIX also change between these two Appendices.

* The Appendices cited under examples refer to the Appendices in Volume 3 of the SENES (1985) report.

SENES, the SYVAC3 code was modified to include corrections for these deficiencies.

The final SYVAC3 results agreed qualitatively with the UTAP results for all 88 variables for both the deterministic simulation and the probabilistic simulations. Although the final agreement was "poor" according to the pre-defined comparison criteria, in many cases the results were in quantitative agreement. For the deterministic comparison, predicted doses due to inhalation of radon and dust, doses due to the ingestion of vegetation, and lake pH values agreed to within 0.5% at all times. The largest differences were less than 20%, and involved the waterborne pathways. For the probabilistic comparisons, a similar pattern was obtained; for example, the waterborne dose pathways again showed the most disagreement, while the terrestrial dose pathways were generally in good agreement.

The code comparison was demonstrated to be an effective tool for identifying ambiguities, inconsistencies, and errors in the documentation of a computer model. Identifying these weaknesses of documentation is important because it is typically the documentation of a computer model, and not the code, that is subject to peer and regulatory review. The presence of accurate documentation is essential for obtaining a fair review.

An action plan has been suggested to eliminate all remaining differences between the SYVAC3 and Phase II version of UTAP. Nevertheless, we do not recommend further work to improve the agreement of SYVAC3 and this version of UTAP, since this comparison has already demonstrated the nature of deficiencies in the documentation of the Phase II version of UTAP. Instead, we recommend the following action plan, inasmuch as the Phase III version of UTAP development is almost complete:

(1) More attention should be given to initial conditions of differential equations and to the physical units of variables in the documentation of the Phase III version of UTAP.

(2) The Phase III version of UTAP should be documented using detailed specifications (cf. Appendices B and C) in addition to English-language and mathematical descriptions.

(3) Further code comparisons should be carried out as independent quality assurance checks whenever UTAP results are to be used for assessments of actual tailings sites, and whenever a high degree of quality assurance is required for any other reason (for example, the Atomic Energy Control Board (AECB, 1985) outlines requirements for quality assurance of codes such as UTAP).

(4) Future code comparisons should be based upon a set of at least six deterministic runs, where all final, intermediate, and sampled variable values are documented, as well as a set of probabilistic runs for which statistical results for the major variables of interest are available. This recommendation would permit an exhaustive examination of all differences, and provide more assurance of the validity of the comparison findings.

ACKNOWLEDGEMENTS

The authors are indebted to a number of people who contributed to this work. They include, from WNRE, B.D. Amiro, K.J. Hoffman, T.W. Melnyk, M.I. Sheppard, G.R. Sherman and M.E. Stephens, who contributed to the walk-throughs; P.A. O'Connor and A. So who contributed to the structured testing and the runs using the Programming Environment package; E.N. Henschell who helped produce the specification figures and A.G. Wikjord who helped polish the draft copy. B.E. Halbert and M. Murray of SENES consultants were instrumental in assisting with many difficulties encountered with the documentation of the tailings model, and in preparing the detailed output from UTAP for the deterministic comparison. R.W. Holmes from the NUTP also provided useful comments for the comparison criteria, and for the contents of this report.

REFERENCES

AECB, 1985. Deep Geological Disposal of Nuclear Fuel Waste: Background Information and Regulatory Requirements Regarding the Concept Assessment Phase. Atomic Energy Control Board Regulatory Document R-71.

Andres, T.H., 1985. Features of the SYVAC3 Executive. In Proceedings of the Twentieth Information Meeting of the Canadian Nuclear Fuel Waste Management Program. Atomic Energy of Canada Limited Technical Record*, TR-375, Pinawa, Manitoba, pp. 365-77.

Deutsch, M.S., 1982. Software Verification and Validation. Prentice-Hall Inc., N.J., 1982.

Dormuth, K.D. and Quick, R.D., 1981. Accounting for Parameter Variability in Risk Assessment for a Canadian Nuclear Fuel Waste Disposal Vault. Int. J. of Energy Systems, 1, pp. 125-127.

Dormuth, K.D. and G.R. Sherman, 1981. SYVAC - A Computer Program for Assessment of Nuclear Fuel Waste Management Systems, Incorporating Parameter Variability. Atomic Energy of Canada Limited Report, AECL-6814, Pinawa, Manitoba.

Dunn, R. and Ullman, R., 1982. Quality Assurance for Computer Software. McGraw-Hill, N.Y., 1982.

Goodwin, B.W., 1984. Evolution of SYVAC3. In Proceedings of the Eighteenth Information Meeting of the Nuclear Fuel Waste Management Program. Atomic Energy of Canada Limited Technical Record*, TR-320, Pinawa, Manitoba, pp. 107-17.

Goodwin, B.W. and E.L.J. Rosinger, 1985. Environmental and Safety Assessment. In Proceedings of the Twentieth Information Meeting of the Canadian Nuclear Fuel Waste Management Program. Atomic Energy of Canada Limited Technical Record*, TR-375, Pinawa, Manitoba, pp. 21-9.

Goodwin, B.W., N.C. Garisto and J.W. Barnard, 1986. An Assessment of the Long-Term Impact of Chemically Toxic Contaminants from the Disposal of Nuclear Fuel Waste. Atomic Energy of Canada Limited Report, AECL-7819 (in preparation).

Guvanassen, V.M., 1985. Preliminary Assessment for the Disposal of Intermediate-Level Reactor Waste. Atomic Energy of Canada Limited Technical Record*, TR-293, Pinawa, Manitoba.

Hamming, R.W., 1971. Introduction to Applied Numerical Analysis. McGraw-Hill, New York.

Harrison, W.C., B.E. Halbert and D.B. Chambers, 1985. Safety Assessment of Uranium Mill Tailings in the National Uranium Tailings Program. Presented at the CNS Conference, Ottawa, May, 1985.

IAEA, 1982. Radioactive Waste Management Glossary. International Atomic Energy Agency publication, IAEA-TECDOC-264, Vienna.

Murray, M.L., D.B. Chambers, D.I. Beals, B.E. Halbert, T.H. Andres and M. Henrion, 1986. Quality Assurance Documentation on the Uranium Tailings Assessment Program. Draft copy of a report prepared for the Canadian National Uranium Tailings Program, and including material from a report by T.H. Andres entitled "Demonstration of Probabilistic Assessment Procedures Using a Radon Dose Model", dated 1985 June.

Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, 1986. Numerical Recipes - The Art of Scientific Computing. Cambridge University Press, Cambridge.

Rummery, T.R. and Rosinger, E.L.J., 1984. The Canadian Nuclear Fuel Waste Management Program. In Fuel Reprocessing and Waste Management, Proceedings of the American Nuclear Society International Topical Meeting, Jackson, Wyoming, 1984 August 26-29, American Nuclear Society, Lagrange Park, Illinois, pp. 1-14 to 1-32.

SENES, 1985. Probabilistic Model Development for the Assessment of the Long-Term Effects of Uranium Mill Tailings in Canada - Phase II. Volume 1 - Main Report. Volume 2 - Workshop Notes and Topical Discussion Papers. Volume 3 - Probabilistic Analysis User's Manual and Documentation for the Uranium Tailings Assessment Program (UTAP).

Sherman, G.R., D.C. Donahue, S.G. King and A. So, 1986. SYVAC2 - A Systems Variability Analysis Code for Assessment of Nuclear Fuel Waste Disposal. Atomic Energy of Canada Limited Technical Record*, TR-317, Pinawa, Manitoba.

Softool, 1984. Programming Environment: Auditor Manual. Softool Corporation, 1984.

Webster, 1985. Webster's New Collegiate Dictionary. Allen & Son Limited, Markham, Ontario.

Wuschke, D.M., A.M. Rice and P.A. Gillespie, 1983. Environmental Assessment of Sub-Seabed Disposal of Nuclear Wastes: A Demonstration Probabilistic Systems Analysis. Atomic Energy of Canada Limited Technical Record*, TR-206, Pinawa, Manitoba.

Wuschke, D.M., P.A. Gillespie, K.K. Mehta, W.F. Heinrich, D.M. LeNeveu, G.R. Sherman, V.M. Guvanasen, D.C. Donahue, B.W. Goodwin, T.H. Andres and R.B. Lyon, 1985. Second Interm Assessment of the Canadian Concept for Nuclear Fuel Waste Disposal. Volume 4: Post-Closure Assessment. Atomic Energy of Canada Limited Report, AECL-8373-4, Pinawa, Manitoba

Wuschke, D.M., K.K. Mehta, K.W. Dormuth, T.H. Andres, G.R. Sherman, E.L.J. Rosinger, B.W. Goodwin, J.A.K. Reid and R.B. Lyon, 1981. Environmental and Safety Assessment Studies for Nuclear Fuel Waste Management. Volume 3: Post-Closure Assessment. Atomic Energy of Canada Limited Technical Record*, TR-127-3, Pinawa, Manitoba.

* Unrestricted, unpublished report, available from SDDO, Atomic Energy of Canada Limited Research Company, Chalk River, Ontario, KOJ 1JO.

Volume

2

NATIONAL
URANIUM
TAILINGS
PROGRAM

Energy, Mines and Resources Canada
Énergie, Mines et Ressources Canada

Canada

COMPARISON OF THE PROBABILISTIC SYSTEMS ASSESSMENT CODES, UTAP AND SYVAC3, APPLIED TO THE LONG-TERM ASSESSMENT OF URANIUM MILL TAILINGS IN CANADA

DSS File #15SQ.23317-5-1720
Contract #OSQ85-00293

VOLUME II: APPENDICES

A Research Report Prepared for:

The National Uranium Tailings Program
Canada Centre for Mineral and Energy Technology
Energy, Mines and Resources Canada
555 Booth Street
Ottawa, Ontario
K1A 0G1

by

B.W. Goodwin and T.H. Andres
Atomic Energy of Canada Research Company
Whiteshell Nuclear Research Establishment
Pinawa, Manitoba
R0E 1L0

1986 March

354pp

MICROMEDIA

ABSTRACT

This report describes a comparison of the output of two codes: UTAP (Uranium Tailings Assessment Program) and SYVAC3 (Systems Variability Analysis Code, generation 3). Both codes are probabilistic systems assessment codes, written to serve as tools in the preparation of long-term environmental assessments. Both codes consist of a central core of 'executive' modules, to which are linked one or more submodels that are representative of the physical system being assessed.

UTAP has been used in a preliminary environmental assessment of the impact of a generic uranium mine tailings site.[1] Available documentation for this assessment includes a description of the model representing the tailings site. From these written descriptions, a set of model-related subroutines were written and linked with SYVAC3. SYVAC3 was then run, using the supplied data, to generate suitable output for comparison with the output from UTAP. This report provides details on the steps used to prepare the code for SYVAC3, and a description of the comparison of the results from UTAP and SYVAC3

NOTICE

This report was prepared by Atomic Energy of Canada Limited (AECL) under contract to the National Uranium Tailings Program (NUTP). Neither the NUTP nor AECL makes any warranty, expressed or implied, or assumes any liability with respect to any damage or loss incurred as a result of the use made of the information in this publication.

1/ SENES. 1985. Probabilistic Model Development for the Assessment of the Long-term Effects of Uranium Mill Tailings in Canada - Phase II. Volume 1 - Main Report. Volume 2 - Workshop Notes and Topical Discussion Papers. Volume 3 - Probabilistic Analysis User's Manual and Documentation for the Uranium Tailings Assessment Program (UTAP)

TABLE OF CONTENTS

	Page
ABSTRACT	i
NOTICE	i
APPENDIX A. ANCILLARY DOCUMENTATION FROM SENES	A-1
APPENDIX B. SPECIFICATIONS FOR THE TAILINGS MODEL	B-1
APPENDIX C. DATA DICTIONARY FOR THE TAILINGS MODEL	C-1
APPENDIX D. RECORDS OF THE QUALITY ASSURANCE WALKTHROUGHS FOR THE SPECIFICATIONS	D-1
APPENDIX E. SAMPLE OUTPUT FROM SOFTOOL'S PROGRAMMING ENVIRONMENT AUDITOR	E-1
APPENDIX F. SAMPLE RESULTS FROM STRUCTURED BASIS MODULE TESTING AND FUNCTIONAL TESTING	F-1
APPENDIX G. DETAILED OUTPUT FROM SYVAC3	G-1
APPENDIX H. LISTINGS OF CODE DEVELOPED FOR THE URANIUM TAILINGS MODEL AS IMPLEMENTED IN SYVAC3	H-1

APPENDIX A. ANCILLARY DOCUMENTATION FROM SENES

During the specification, design and data preparation phases, some problems and discrepancies were encountered in extracting required information from the material in the SENES report (SENES, 1985). A number of telephone conversations and letters between SENES and the Whiteshell Nuclear Research Establishment were required to resolve these difficulties.

Since the resultant information forms a significant part of the work described in this report, a full record was kept of the questions passed to SENES and the answers which SENES provided. The following pages are reproductions of letters from SENES and of notes taken by B.W. Goodwin during telephone conversations with SENES. They are arranged in chronological order, and include:

			Page
1.	Notes on a call to SENES	Feb. 6, 1986	A-2
2.	Notes on a return call from SENES	Feb. 10, 1986	A-3
3.	Letter from SENES	Feb. 12, 1986	A-5
4.	Letter from SENES	Feb. 13, 1986	A-11
5.	Notes on a call to SENES	Mar. 5, 1986	A-14
6.	Notes on a call to SENES	Mar. 11, 1986	A-15
7.	Letter from SENES	Mar. 17, 1986	A-16
8.	Notes on a call to SENES	Apr. 8, 1986	A-36
9.	Notes on a return call from SENES	Apr. 9, 1986	A-37
10.	Notes on a call to SENES	Apr. 10, 1986	A-38
11.	Notes on a return call from SENES	Apr. 10, 1986	A-39
12.	Letter from SENES	Apr. 14, 1986	A-40
13.	Notes on a call to SENES	May 6, 1986	A-48
14.	Notes on a return call from SENES	May 6, 1986	A-49

Call to SENES Feb. 6/86 3 p.m.

Doug Chambers & Bruce Halbert

1. Table 5.1 or Tables in Appendices for data?

2. P. A-15 e^{-kt_h} should be e^{-kt}?

3. Eq. (A.3.20): t should be t_h?

4. Eq. (A.3.21): g_oh should be $g_o\lambda_w$?

5. $C_{T,t+\Delta t} = C_T(t+\Delta t,\lambda_w)$ (from Eq. (A.3.17))? (B.H. concurred).

6. Before Eq. (A.3.23) should be ... + [radium leaving with porewater in Δt]?

7. P A.19 How derived?

8. Eq. (A.3.24): what is $x_{t=0}$?

9. Units: $[Ra_s]_1$ in 6.1 etc. are $[\text{mol m}^{-3}]$
   in 1.1 $[\text{Bq m}^{-3}]$
   Table B.1 $[\text{Bq g}^{-1}]$
   Which is correct?

10. Is $[\overline{Ra}]_{\lambda_w}$ = Eq. (A.3.2) + Eq. (A.5.5)?

11. In Eq. (A.7.9) and (A.7.10), should λ_{mix} be d_{mix}?

12. Is Eq. (A.7.1) correct?

13. "+" sign missing in Eq. (A.7.13)?

14. Is $k_2 = \beta k_1$ in Appendix C? (B.H. concurred).

15. Does CAIR involve only Ra to vegetation?

16. Geosphere transport is not in UTAP? (B.H. concurred).

17. What are units of constants in Eq. (B.1.1)? (B.H. referred to p. B-2).

18. See fixes to pH Eq'ns (A.4.1) to come from B.H.

Return Call from SENES 10/2/86 8:30 a.m.

Bruce Halbert

1. Ignore the data in Table 5.1. Use the data in the Appendices with some changes noted below.

2. Wind speeds etc. in Appendices are annual means, whereas the values to be used should be adjusted to the time period of the simulation (500a). Bruce will send a corrected list.

3. P 4-15: as noted, e^{-kt_h} should be e^{-kt}(twice)

4. P 4-16: as noted, RHS of Eq. (A.3.21) should be $g_o\lambda_w$ (not $g_o h$)

5. As noted, C_T, $C_{T,t+\Delta t}$ etc. should be $C_T(t+\Delta t, \lambda_w)$ as defined in Eq. (A.3.17) and used many places.

6. P A-18: as noted, central equation (above A.3.23) should be ... + [radium leaving with porewater in Δt]

7. P A.19: Derivation of equation at top will be sent from Jano.

8. Correction to pH equation (A.4.1) - RHS:

$$\frac{(C_s + C_T)\,[2 - f^{3/4}\,(10^{9.7-3pH}) - f^3\,(10^{8.99-3pH})]}{1 + f^{3/4}\,(10^{9.7-3pH}) + f^3(10^{8.59-3pH})} + \ldots$$

9. P A-28 Correction to Eq. (A.7.6):

$$\ldots + \frac{f^3\,(10^{8.99-3pH})\ \ldots}{1 + f^3\,(\ ) + f^{3/4}(\ )}$$

10. P A-29 Correction to Eq. (A.7.8):

$$\ldots\ \frac{2 - f^{3/4}\ 10^{10-3pH}}{1 + f^{3/4}\ 10^{10-3pH}} + \ldots$$

11. Eq. (A.7.9), (A.7.10) and (A.7.11): λ_{mix} should be d_{mix}, as noted.

12. $[\overline{Ra}]_{\lambda_w} = [Ra]^{gyp}_{aq} + [Ra]^{Th}_{aq} =$ Eq. (A.3.2) + Eq. (A.5.5), as noted.

13. Eq. A.7.11 should be

$$[Ra]_{t+\Delta t} = [\overline{Ra}]_{\lambda_w} + \left\{[Ra]_t - [\overline{Ra}]_{\lambda_w}\right\} \exp\left[- \frac{P\Delta t}{\varepsilon d_{mix}(1+K_{D,Ra}\rho_T/\varepsilon)}\right]$$

14. Eq. (A.7.13) is OK - no + sign as was noted.

15. Table A.1 X_{gyp} = FRACGP.Data should be:
Constant (0.05)

16. Eq. (B.1.1) C_i should be in [Bq m^{-3}]: convert RACON1.
For eq. (B.2.4), S_i is proportional to $(Ra)_i$. For use later,
$(Ra)_i$ should be in [Bq m-3], so delete ρ_T from equation.
Use [Bq m^{-3}] in Tables B.1 and B.2.

17. Lake model: $k_2 = \beta k_1$

18. Fix to eq. on P C 12
3 [ ] + 2 [ ] + ( ) + ( )

19. Table C.1: Use the following data: BETA LN (.18, 1.35, .1, 33)
ACTO CONST (0.8)

SENES CONSULTANTS LIMITED

499 McNicoll Ave.
Willowdale, Ontario
Canada M2H 2C9
Tel: (416) 499-5030

30206/30259

12 February 1986

Dr. Bruce Goodwin
Atomic Energy of Canada Limited
Whiteshell Nuclear Research Establishment
Pinawa, MN
ROE 1L0

Re: Questions Pertaining to Our Phase II Documentation on Probabilistic Assessment of Uranium Mill Tailings

Dear Bruce:

Further to your queries of 6 February 1986 and our subsequent telephone discussion of 10 February, 1986, we have prepared the attached commentary to clarify certain points you raised. These questions pertain to the component model documentation given in Volume 3.

Please feel free to call should you have any other questions.

Yours very truly,

SENES CONSULTANTS LIMITED

B.E. Halbert,
Principal

attach.

cc: R. Holmes, NUTP

Attachment to Dr. Bruce Goodwin
12 February 1986

1. Equation (3.24) on page A-19

The derivation of equation (3.24) requires a function to approximate the integral:

$$\int_{t}^{t+\Delta t} P\ C_T y\ dt = PK \int_{t}^{t+\Delta t} C_T x_t dt$$

where both C_T and x_t are a function of time.

If Δt is small, the simplest function is:

$$= PK \left[\frac{C_{T,t+\Delta t}\ x_{t+\Delta t} - C_{T,t}\ x_t}{2} \right] \Delta t$$

where $\left[\frac{C_{T,t+\Delta t}\ x_{t+\Delta t} + C_{T,t}\ x_t}{2} \right]$ is the arithmetic mean between t and $t+\Delta t$.

However, if Δt is large or x_t changes rapidly with time a logarithmic mean is a better approximation (for the interval t and $t+\Delta t$). Let $C_{T,t}x_t$ be an exponentially rising function:

$$C_{T,t}x_t = (C_T x_t)_o e^{\alpha t}$$

where $(C_T x_t)_o$ is an arbitrary initial condition.

Integrating between t and t+Δt gives:

$$\int_{t}^{t+\Delta t} (C_T x_t)_o e^{\alpha t}\, dt = \frac{(C_T x_t)_o e^{\alpha(t+\Delta t)} - (C_T x_t)_o e^{\alpha t}}{\alpha}$$

since $(C_T x_t)_o e^{\alpha t} = C_{T,t} x_t$

and letting $(C_T x_t)_o e^{\alpha(t+\Delta t)} = C_{T,t+\Delta t} x_{t+\Delta t}$

then $\alpha t = \ln(C_{T,t} x_t) - \ln(C_T x_t)_o$

and $\alpha(t+\Delta t) = \ln(C_{T,t+\Delta t} x_{t+\Delta t}) - \ln(C_T x_t)_o$

Eliminating $(C_T x_t)_o$ yields:

$$\alpha \Delta t = \ln(C_{T,t+\Delta t} x_{t+\Delta t}) - \ln(C_{T,t} x_t)$$

or

$$\alpha = \frac{1}{\Delta t} \ln\left[\frac{C_{T,t+\Delta t} x_{t+\Delta t}}{C_{T,t}\, x_t}\right]$$

Thus

$$PK \int_{t}^{t+\Delta t} C_T x_t dt = PK \left[\frac{C_{T,t+\Delta t} x_{t+\Delta t} - C_{T,t} x_t}{\ln\left[\frac{C_{T,t+\Delta t} x_{t+\Delta t}}{C_{T,t} x_t}\right]}\right] \Delta t$$

Note that if the difference between $C_{T,t+\Delta t} x_{t+\Delta t}$ and $C_{T,t} x_t$ is small, say 10%, the logarithmic mean approaches the arithmetic mean.

Letting $C_{T,t+\Delta t} x_{t+\Delta t} = 0.11$

and $C_{T,t} x_t = 0.10$

then $\dfrac{0.11-0.10}{\ln\left(\dfrac{0.11}{0.10}\right)} = 0.104921$ versus 0.105 for the arithmetic mean.

We have found that the radium content in the porewater rises rapidly (essentially an exponential function) as the gypsum content in the porewater is depleted. When the gypsum content is still high, C_Tx_t changes slowly, it makes little difference which approach one uses. Therefore, we have decided to use the log mean concept throughout.

2. Equation (7.11) on Page A-30

Equation (7.11) pertains to the estimation of the dissolved radium concentration in the porewater of the mixing zone. Development of the equation is documented below. Let

V_T = total volume of the mixing zone (m^3)
V_ℓ = volume occupied by liquid (m^3)
ρ = bulk density (mass solid/V_T) of the tailings ($g.m^{-3}$)
$[Ra]_{mix}$ = concentration of radium in the liquid phase ($mol.m^{-3}$)

The mass of radium in the liquid phase at any time "t" is then given by:

$$V_\ell[Ra]_{mix}$$

and the mass of radium per unit total tailings volume

$$= \frac{V_\ell}{V_T}[Ra]_{mix}$$

Also the mass of radium adsorbed on tailings solids per unit total tailings volume is given by:

$$\rho K_D[Ra]_{mix}$$

where K_D = the distribution coefficient for radium between the solid phase and liquid phase ($m^3.g^{-1}$).

Material balance yields:

$$\frac{d}{dt} V_T \left[\rho K_D[Ra]_{mix} + \frac{V_\ell}{V_T} [Ra]_{mix} \right] = F[\overline{Ra}]_{\lambda w} - F[Ra]_{mix}$$

where

F = flow rate of water from the unsaturated zone ($m^3.a^{-1}$)

$[\overline{Ra}]\lambda_w$ = the average concentration of radium entering the mixing zone between t and $t+\Delta t$ ($mol.m^{-3}$)

On defining the following parameters:

P = net precipitation on the tailings surface ($m.a^{-1}$)

A = area of the mixing zone (m^2)

d_{mix} = thickness of the mixing zone (m)

$$\epsilon = \frac{V_\ell}{V_T}$$

$$F = PA$$

and

$$V_T = d_{mix} A$$

the material balance may be restated as:

$$d_{mix}\,\epsilon \left[1 + \frac{\rho K_D}{\epsilon} \right] \frac{d[Ra]_{mix}}{dt} = P[\overline{Ra}]_{\lambda w} - P[Ra]_{mix}$$

Integration between t and t+Δt yields equation (7.11):

$$[Ra]_{mix,t+\Delta t} = [\overline{Ra}]_{\lambda w} + \left[[Ra]_{mix,t} - [\overline{Ra}]_{\lambda w}\right] e^{\left\{\frac{-P\Delta t}{d_{mix}\epsilon\left[\frac{1+\rho K_D}{\epsilon}\right]}\right\}}$$

The value $[\overline{Ra}]\lambda_w$ is time dependent as long as the gypsum is not depleted from the unsaturated zone. Also the gypsum in the mixing zone cannot dissolve if the porewater entering is already saturated. After the depletion of the gypsum in the unsaturated zone, $[\overline{Ra}]\lambda_w$ can be regarded as a constant, depending upon radium ingrowth. Subsequently the $[Ra]_{mix}$ is affected by the release of radium by gypsum dissolution in the mixing zone. I realize, that that is not strictly true, but otherwise the calculations are complicated.

3. Base Case Parameter Values

Attached hereto is a computer listing of the parameter values used for the base case scenario reported in Phase II. The values are mainly the same as those reported in the tables at the back of each appendix. As I pointed out during our telephone conversation of 10 February 1986, the values given in the appendices for the time-variant parameters (e.g. net precipitation, wind speed, etc.) are distributions for mean annual values. When used in the Phase II model, the variances, minimums and maximums were adjusted for the time period of the simulation (i.e. 500 years in many runs). The values given on the computer listing for these parameters have been adjusted for a 500 year simulation period.

SENES CONSULTANTS LIMITED

499 McNicoll Ave.
Willowdale, Ontario
Canada M2H 2C9
Tel: (416) 499-5030

30206/30259

13 February 1986

Dr. Bruce Goodwin
Atomic Energy of Canada Limited
Whiteshell Nuclear Research Establishment
Pinawa, MN
ROE 1L0

Re: Questions Pertaining to Our Phase II Documentation on Probabilistic Assessment of Uranium Mill Tailings

Dear Bruce:

Please find enclosed the computer listing of input parameter values which was inadvertently left out of the package sent to you yesterday on the above topic.

Yours very truly,

SENES CONSULTANTS LIMITED

B.E.Halbert, M.S.
Principal

sdp

Encl.

cc: R. Holmes, NUTP

— Specialists in Energy, Nuclear and Environmental Sciences —

```
           PROB. ANAL. OF U MILL TAILINGS:BASECASE
   0  97   0   0   0
 0913566091 500
 FWIND1    2        3.5E1      3.0E-1      3.41E01     3.59E01
 FWIND2    2        .1E00      8.0E-4      9.73E-2     1.03E-1
 FPREC     2       1.23E2      8.0E-1      1.21E02      1.25E2
 UWIND     2       5.3E00      2.2E-2       5.2E00      5.4E00
 RYEARS    1        5.0E2
 DT        1       1.0E01
 OTIME     1       5.0E01
 CODS      1       0.0E00
 CODL      1       0.0E00
 CODPH     1       0.0E00
 CODF      1       0.0E00
 CODR      1       0.0E00
 CODD      1       0.0E00
 CODA      1       0.0E00
 CODV1     1       0.0E00
 CODV2     1       0.0E00
 CODO      1       0.0E00

 CODU      1       0.0E00
 SULPRD    3       1.4E-6       1.5E0      0.4E-6       4.2E-6
 RHOTAL    2        1.4E6       0.1E6      1.1E06       1.7E06
 O2CONC    1         9.04
 O2DIFF    6         30.0        70.0       130.0
 KPRIME    3       5.0E-2         1.7        0.02         0.25
 KSUBSO    1        2.5E1
 NPRCP     2       4.0E-1      4.0E-3      3.9E-1       4.1E-1
 FACTCO    6       2.0E-1      3.0E-1      5.0E-1
 DISTCO    4       1.0E-2      1.0E-1
 DUNSAT    1          1.0
 DMIX      1          1.0
 KSUBDR    3      80.0E-6      3.5E00      1.9E-6       3.4E-3
 RAINGP    6       5.0E-2      2.0E-1      6.0E-1
 TVOID     6       3.5E-1      4.0E-1      4.5E-1
 CONCRA    6          5.0       1.1E1       1.5E1
 FRACGP    1       5.0E-2
 PYRITE    1        7.3E2
 UPERTH    1       3.0E00
 UO2TON    1       1.2E00
 RAINTH    6       8.5E-1      9.0E-1      9.5E-1
 THDISS    4       1.0E-2      3.0E-2
 THFRAC    6          0.2      3.0E-1         0.4
 XK1       3       4.3E-1     1.85E00     1.26E-1      1.47E00
 XK3       3       2.2E-2     1.20E00     1.53E-2      3.17E-2
 BETA      3       1.8E-1     1.35E00     1.00E-1      3.28E-1
```

```
ALAKE     1     1.6E06
LAKE      1     1.0E01
AWAT      1     2.0E07
ATAIL     1     5.0E05
CTCO      1     0.8E00
F         3     1.60E-5    3.50E00    1.0E-6    6.0E-4
MPCH2O    6     5.0E00     1.5E01     2.5E01
NLAM      1     2.1E-6
MANAT     6     2.0E-2     2.0E-1     9.0E-1
SILT      1      .3E2
AEXPT     1      5.0E5
SRC       4     1.0E00     3.0E00
XDIST     1     3.6E03
VDISP     3     6.0E01     1.08E0     4.76E1    7.57E1
DEPSF1    4     8.0E-1     1.0E00
DEPSF2    2     4.7E-1     3.0E-1     1.0E-2    1.0E00
FDEPSF3   4     1.0E-1     3.0E-1
DEPSF4    4     1.0E-1     3.0E-1
DEPED1    4     8.0E-1     1.0E00
FDEPED2   4     8.0E-1     1.0E00
FDEPED3   4     8.0E-1     1.0E00
DEPED4    4     5.0E-2     1.5E-1
DUREXP1   4     9.0E07     1.1E08
DUREXP2   4     2.7E06     3.1E06
UREXP3    4     4.7E06     5.6E06
UREXP4    4     4.7E06     5.6E06
DEPVEL    4     3.0E-3     3.0E-2
EGLAM1    4     2.0E-9     4.0E-9
EGLAM2    4     4.0E-7     6.0E-7
VEGLAM3   4     4.0E-7     6.0E-7
VEGLAM4   4     4.0E-7     6.0E-7
DEN1      4     4.0E02     6.0E02
YDEN2     4     5.0E02     1.0E03
YDEN3     4     1.5E03     2.5E03
DEN4      4     1.5E03     2.5E03

FBI       4      .30E-9    2.5E-9
AUT1      6     0.50E-1    0.2E00     .50E00
AUT2      6     0.50E-1    0.2E00     .50E00
FIR       1     2.5E03

CFING     1   3.200E-7
VEGINA    1   1.570E05
VEGINB    1     7.0E04
ACTA      4     5.0E-1      1.0E0
FACTB     4     5.0E-1      1.0E0
FLOCA     6      .1E00      .30E0      .6E00
MEATI     1     7.9E04
MEAT      4     0.0E00      1.0E0
WATIN     1      .6E00
ISHIN     4     4.0E03     1.2E04
FISH      1     1.0E00
DCFINH    1     2.6E-6
INHIN     1     8.4E03
F         6     4.0E-1     6.0E-1     9.0E-1
F         6     2.0E-1     4.0E-1     7.0E-1
```

Call to SENES 1986 March 5

(Bruce Halbert)

1. Is the data for FIR correct? Was 10^3 higher in Table 5.1 of Vol.3. In the letter of February 13/86 it is 2.5×10^3 g(net weight)a^{-1}

2. Eq. (A.3.24): What is $x_{t=0}$?

3. Bruce noted that Ron Holmes asked SENES to send us the "latest" results from UTAP, incorporating some errors that have been detected, such as the error in the calculation of lake pH. Also Ron Holmes asked SENES to send us a single deterministic run.

Call to SENES March 11/86 2:11 p.m.
(Bruce Halbert)

1. Initial condition is x=0 at t=0

2. Caribou data: was run with an incorrect value FIR = (to be confirmed)

3. Bruce Halbert will provide us with a single deterministic run for comparison. Probably with the correct value for FIR.

4. We shall attempt to reproduce the probabilistic results in the SENES report.

SENES CONSULTANTS LIMITED

499 McNicoll Ave.
Willowdale, Ontario
Canada M2H 2C9
Tel; (416) 499-5030

30259

17 March 1986

Dr. Bruce Goodwin
Atomic Energy of Canada Ltd.
Whiteshell Nuclear Research Estab.
Pinawa, Manitoba.
ROE 1L0

Re: Phase II Probabilistic Results for the Uranium Tailings Assessment Program

Dear Bruce:

Further to our recent conversations, I am enclosing several sets of our probabilistic analysis results which may be useful for comparison purposes. The results were all generated using a modified version of the Phase II code. As we discussed previously, the modifications involved some of the pH calculations and the calculation of dissolved ferric iron. None of these changes however, affect pH levels predicted for the lake water column. While we acknowledge that one of the input parameters (FIR) was incorrect by a factor of 1000; we have not made this change to the input data set.

Tables 1 and 2 present the input distributions and output results respectively for 500 probabilistic trials of the modified program. Tables 3 and 4 present nominal value inputs and the results of a deterministic trial. These deterministic results may be quite useful for initial comparison of our respective program results. Table 5 is a key to the output result variable names given on Tables 2 & 4.

.../2

I hope you find the enclosed material useful. Please call should you have any questions.

Yours very truly,

SENES CONSULTANTS LIMITED

Bruce

Bruce E. Halbert,
Prinicipal

ls

Encl.

cc. R. Holmes, NUTP

TABLE 1

```
        PROB. ANAL. OF U MILL TAILINGS:BASECASE
  0  97   0   0   0
0913566091 500
 FWIND1   2      3.5E1       3.0E-1      3.41E01      3.59E01
 FWIND2   2      .1E00       8.0E-4      9.73E-2      1.03E-1
 FPREC    2     1.23E2       8.0E-1      1.21E02       1.25E2
 UWIND    2     5.3E00       2.2E-2       5.2E00       5.4E00
 RYEARS   1      5.0E2
 DT       1     1.0E01
 OTIME    1     5.0E01
 CODS     1     0.0E00
 CODL     1     0.0E00
 CODPH    1     0.0E00
 CODF     1     0.0E00
 CODR     1     0.0E00
 CODD     1     0.0E00
 CODA     1     0.0E00
 CODV1    1     0.0E00
 CODV2    1     0.0E00
 CODO     1     0.0E00

 CODU     1     0.0E00
 SULPRD   3     1.4E-6        1.5E0       0.4E-6       4.2E-6
 RHOTAL   2      1.4E6        0.1E6       1.1E06       1.7E06
 O2CONC   1       9.04
 O2DIFF   6       30.0         70.0        130.0
 KPRIME   3     5.0E-2          1.7         0.02         0.25
 KSUBSO   1      2.5E1
 NPRCP    2     4.0E-1       4.0E-3       3.9E-1       4.1E-1
 FACTCO   6     2.0E-1       3.0E-1       5.0E-1
 DISTCO   4     1.0E-2       1.0E-1
 DUNSAT   1        1.0
 DMIX     1        1.0
 KSUBDR   3    80.0E-6       3.5E00       1.9E-6       3.4E-3
 RAINGP   6     5.0E-2       2.0E-1       6.0E-1
 TVOID    6     3.5E-1       4.0E-1       4.5E-1
 CONCRA   6        5.0        1.1E1        1.5E1
 FRACGP   1     5.0E-2
 PYRITE   1      7.3E2
 UPERTH   1     3.0E00
 UO2TON   1     1.2E00
 RAINTH   6     8.5E-1       9.0E-1       9.5E-1
 THDISS   4     1.0E-2       3.0E-2
 THFRAC   6        0.2       3.0E-1          0.4
 XK1      3     4.3E-1     1.85E00      1.26E-1      1.47E00
 XK3      3     2.2E-2     1.20E00      1.53E-2      3.17E-2
 BETA     3     1.8E-1     1.35E00      1.00E-1      3.28E-1

 ALAKE    1     1.6E06
 DLAKE    1     1.0E01
 AWAT     1     2.0E07
 ATAIL    1     5.0E05
 ACTCO    1     0.8E00
 BF       3   1.60E-5     3.50E00       1.0E-6       6.0E-4
 MPCH2O   6     5.0E00       1.5E01       2.5E01
 RNLAM    1     2.1E-6
 EMANAT   6     2.0E-2       2.0E-1       9.0E-1
```

```
ILT       1      .3E2
AEXPT     1     5.0E5
HSRC      4    1.0E00      3.0E00
DIST      1    3.6E03
VDISP     3    6.0E01      1.08E0      4.76E1      7.57E1
FDEPSF1   4    8.0E-1      1.0E00
DEPSF2    2    4.7E-1      3.0E-1      1.0E-2      1.0E00
DEPSF3    4    1.0E-1      3.0E-1
FDEPSF4   4    1.0E-1      3.0E-1
FDEPED1   4    8.0E-1      1.0E00
DEPED2    4    8.0E-1      1.0E00
FDEPED3   4    8.0E-1      1.0E00
FDEPED4   4    5.0E-2      1.5E-1
UREXP1    4    9.0E07      1.1E08
UREXP2    4    2.7E06      3.1E06
DUREXP3   4    4.7E06      5.6E06
UREXP4    4    4.7E06      5.6E06
EPVEL     4    3.0E-3      3.0E-2
VEGLAM1   4    2.0E-9      4.0E-9
VEGLAM2   4    4.0E-7      6.0E-7
EGLAM3    4    4.0E-7      6.0E-7
VEGLAM4   4    4.0E-7      6.0E-7
YDEN1     4    4.0E02      6.0E02
DEN2      4    5.0E02      1.0E03
DEN3      4    1.5E03      2.5E03
YDEN4     4    1.5E03      2.5E03

BI        4    .30E-9      2.5E-9
FAUT1     6   0.50E-1      0.2E00     .50E00
FAUT2     6   0.50E-1      0.2E00     .50E00
IR        1    2.5E03

DCFING    1  3.200E-7
EGINA     1  1.570E05
EGINB     1    7.0E04
FACTA     4    5.0E-1       1.0E0
ACTB      4    5.0E-1       1.0E0
LOCA      6     .1E00       .30E0      .6E00
XMEATI    1    7.9E04
FMEAT     4    0.0E00       1.0E0
ATIN      1     .6E00
FISHIN    4    4.0E03      1.2E04
FFISH     1    1.0E00
FINH      1    2.6E-6
NHIN      1    8.4E03
OF        6    4.0E-1      6.0E-1     9.0E-1
          6    2.0E-1      4.0E-1     7.0E-1
```

TABLE 2

MAR. 13
500 TRIAL
corrected progr
BASECASE.DAT

```
TYPE STATS

DV0     .2618E+01   .1923E+01   .1498E+01   .6208E+01   .1987E+01   .2199E+01   .1884E+00   .1284E+02

DV50    .2583E+01   .1897E+01   .1500E+01   .6225E+01   .1960E+01   .2198E+01   .1864E+00   .1269E+02

DV100   .2548E+01   .1871E+01   .1503E+01   .6242E+01   .1935E+01   .2197E+01   .1843E+00   .1253E+02

DV150   .2514E+01   .1846E+01   .1505E+01   .6260E+01   .1909E+01   .2197E+01   .1823E+00   .1237E+02

DV200   .2481E+01   .1822E+01   .1507E+01   .6278E+01   .1885E+01   .2196E+01   .1804E+00   .1222E+02

DV250   .2449E+01   .1797E+01   .1510E+01   .6296E+01   .1861E+01   .2195E+01   .1784E+00   .1208E+02

DV300   .2417E+01   .1774E+01   .1512E+01   .6314E+01   .1837E+01   .2194E+01   .1766E+00   .1193E+02

DV350   .2386E+01   .1751E+01   .1515E+01   .6332E+01   .1814E+01   .2193E+01   .1747E+00   .1179E+02

DV400   .2356E+01   .1728E+01   .1517E+01   .6351E+01   .1791E+01   .2192E+01   .1729E+00   .1166E+02

DV450   .2326E+01   .1706E+01   .1520E+01   .6369E+01   .1769E+01   .2191E+01   .1712E+00   .1152E+02

DV500   .2297E+01   .1685E+01   .1522E+01   .6388E+01   .1747E+01   .2190E+01   .1694E+00   .1139E+02

DM0     .1852E-02   .2247E-02   .2783E+01   .1425E+02   .9059E-03   .4011E+01   .3427E-05   .1749E-01

DM50    .1827E-02   .2215E-02   .2778E+01   .1420E+02   .8939E-03   .4010E+01   .3378E-05   .1723E-01

DM100   .1802E-02   .2183E-02   .2772E+01   .1416E+02   .8822E-03   .4009E+01   .3331E-05   .1695E-01

DM150   .1778E-02   .2153E-02   .2766E+01   .1411E+02   .8707E-03   .4008E+01   .3284E-05   .1674E-01

DM200   .1755E-02   .2123E-02   .2761E+01   .1406E+02   .8594E-03   .4006E+01   .3239E-05   .1650E-01

DM250   .1732E-02   .2093E-02   .2755E+01   .1401E+02   .8484E-03   .4005E+01   .3194E-05   .1626E-01

DM300   .1709E-02   .2064E-02   .2749E+01   .1396E+02   .8376E-03   .4004E+01   .3150E-05   .1603E-01

DM350   .1687E-02   .2036E-02   .2744E+01   .1391E+02   .8270E-03   .4003E+01   .3108E-05   .1580E-01

DM400   .1665E-02   .2009E-02   .2738E+01   .1386E+02   .8167E-03   .4002E+01   .3066E-05   .1558E-01

DM450   .1644E-02   .1982E-02   .2732E+01   .1381E+02   .8065E-03   .4001E+01   .3025E-05   .1537E-01

DM500   .1623E-02   .1956E-02   .2727E+01   .1376E+02   .7966E-03   .4000E+01   .2985E-05   .1516E-01

DW0     .5608E+01   .2676E+01   .1340E+01   .5433E+01   .5057E+01   .1577E+01   .1307E+01   .1837E+02

DW50    .8322E+01   .5222E+01   .2655E+01   .1405E+02   .7210E+01   .1677E+01   .1606E+01   .4530E+02

DW100   .1855E+02   .2823E+02   .5936E+01   .4767E+02   .1214E+02   .2251E+01   .2228E+01   .2925E+03

DW150   .4278E+02   .5261E+02   .3367E+01   .1846E+02   .2619E+02   .2632E+01   .2407E+01   .4415E+03

DW200   .3873E+02   .4163E+02   .3210E+01   .1644E+02   .2644E+02   .2357E+01   .2271E+01   .3015E+03

DW250   .2706E+02   .2405E+02   .2635E+01   .1409E+02   .1972E+02   .2271E+01   .9515E+00   .2032E+03
```

```
DW350   .1609E+02   .1276E+02   .1839E+01   .8205E+01   .1164E+02   .2421E+01   .5868E+00   .9237E+02

400     .1330E+02   .1047E+02   .1574E+01   .6545E+01   .9462E+01   .2495E+01   .5467E+00   .6442E+02

W450    .1130E+02   .8964E+01   .1460E+01   .6042E+01   .7912E+01   .2545E+01   .5396E+00   .5788E+02

DW500   .9799E+01   .7897E+01   .1442E+01   .6133E+01   .6767E+01   .2578E+01   .5383E+00   .5402E+02

0       .3944E+01   .5764E+01   .3215E+01   .1645E+02   .1782E+01   .3708E+01   .6485E-01   .4295E+02

DF50    .5748E+01   .8511E+01   .3099E+01   .1507E+02   .2540E+01   .3760E+01   .8512E-01   .6069E+02

100     .1258E+02   .2702E+02   .5179E+01   .3634E+02   .4279E+01   .4296E+01   .9217E-01   .2753E+03

F150    .2812E+02   .5585E+02   .5135E+01   .4345E+02   .9227E+01   .4649E+01   .1213E+00   .6671E+03

DF200   .2536E+02   .4524E+02   .3993E+01   .2335E+02   .9316E+01   .4351E+01   .9055E-01   .3917E+03

250     .1835E+02   .3305E+02   .4495E+01   .2963E+02   .6950E+01   .4294E+01   .3795E-01   .2903E+03

DF300   .1421E+02   .2729E+02   .5268E+01   .4013E+02   .5235E+01   .4406E+01   .3603E-01   .2723E+03

350     .1160E+02   .2344E+02   .5652E+01   .4933E+02   .4102E+01   .4559E+01   .2572E-01   .2552E+03

DF400   .9756E+01   .2053E+02   .6361E+01   .5819E+02   .3334E+01   .4676E+01   .2183E-01   .2392E+03

DF450   .8385E+01   .1824E+02   .6808E+01   .6659E+02   .2787E+01   .4760E+01   .2119E-01   .2243E+03

500     .7326E+01   .1637E+02   .7198E+01   .7439E+02   .2384E+01   .4820E+01   .2109E-01   .2104E+03

DD0     .9938E+00   .3203E+00   .5338E+00   .3152E+01   .9421E+00   .1395E+01   .2987E+00   .2115E+01

50      .9803E+00   .3152E+00   .5343E+00   .3153E+01   .9296E+00   .1394E+01   .2947E+00   .2087E+01

DD100   .9672E+00   .3103E+00   .5348E+00   .3155E+01   .9174E+00   .1393E+01   .2909E+00   .2059E+01

DD150   .9544E+00   .3055E+00   .5354E+00   .3156E+01   .9055E+00   .1392E+01   .2871E+00   .2032E+01

200     .9418E+00   .3009E+00   .5359E+00   .3158E+01   .8938E+00   .1391E+01   .2835E+00   .2006E+01

DD250   .9295E+00   .2963E+00   .5365E+00   .3160E+01   .8823E+00   .1389E+01   .2799E+00   .1980E+01

300     .9174E+00   .2918E+00   .5371E+00   .3161E+01   .8711E+00   .1388E+01   .2763E+00   .1954E+01

DD350   .9056E+00   .2874E+00   .5377E+00   .3163E+01   .8601E+00   .1387E+01   .2729E+00   .1929E+01

DD400   .8941E+00   .2832E+00   .5383E+00   .3165E+01   .8493E+00   .1386E+01   .2695E+00   .1905E+01

450     .8828E+00   .2790E+00   .5389E+00   .3168E+01   .8388E+00   .1385E+01   .2662E+00   .1881E+01

DD500   .8717E+00   .2749E+00   .5395E+00   .3170E+01   .8285E+00   .1384E+01   .2630E+00   .1858E+01

0       .1080E+02   .9785E+01   .2210E+01   .9915E+01   .7586E+01   .2383E+01   .5755E+00   .7128E+02

DR50    .9496E+01   .9213E+01   .2383E+01   .1100E+02   .6404E+01   .2501E+01   .4280E+00   .6930E+02

R100    .8904E-01   .8798E+01   .2459E+01   .1155E+02   .5965E+01   .2513E+01   .4102E+00   .6779E+02

150     .7950E+01   .7671E+01   .2569E+01   .1331E+02   .5425E+01   .2459E+01   .4027E+00   .6639E+02

DR200   .7520E+01   .7237E+01   .2640E-01   .1415E+02   .5163E+01   .2437E+01   .3971E+00   .6466E+02

250     .7338E+01   .7003E+01   .2399E+01   .1125E+02   .5086E+01   .2432E+01   .3917E+00   .5526E+02

DR300   .7291E+01   .6907E+01   .2398E+01   .1125E+02   .5022E+01   .2431E+01   .3864E+00   .5453E+02
```

```
DR350   .7197E+01  .6614E+01  .2397E+01  .1125E+02  .4955E+01  .2430E+01  .3812E+00  .5382E+02
DR400   .7104E+01  .6723E+01  .2396E+01  .1125E+02  .4896E+01  .2430E+01  .3761E+00  .5312E+02
DR450   .7014E+01  .6633E+01  .2395E+01  .1125E+02  .4835E+01  .2429E+01  .3711E+00  .5244E+02
DR500   .6926E+01  .6546E+01  .2394E+01  .1125E+02  .4776E+01  .2428E+01  .3662E+00  .5177E+02
TD0     .2397E+02  .1240E+02  .1416E+01  .5643E+01  .2122E+02  .1642E+01  .4289E+01  .8086E+02
TD50    .2713E+02  .1455E+02  .1365E+01  .5039E+01  .2383E+02  .1666E+01  .4593E+01  .8792E+02
TD100   .4355E+02  .4941E+02  .5471E+01  .4643E+02  .3263E+02  .1991E+01  .6447E+01  .5768E+03
TD150   .8233E+02  .9105E+02  .3214E+01  .1735E+02  .5600E+02  .2309E+01  .8157E+01  .7801E+03
TD200   .7503E+02  .7449E+02  .3316E+01  .1883E+02  .5502E+02  .2125E+01  .5556E+01  .6851E+03
TD250   .5618E+02  .4787E+02  .2922E+01  .1502E+02  .4372E+02  .1995E+01  .4138E+01  .3722E+03
TD300   .4507E+02  .3711E+02  .3321E+01  .1939E+02  .3580E+02  .1948E+01  .3445E+01  .3141E+03
TD350   .3818E+02  .3100E+02  .3601E+01  .2331E+02  .3056E+02  .1930E+01  .3290E+01  .2845E+03
TD400   .3342E+02  .2684E+02  .3826E+01  .2696E+02  .2688E+02  .1920E+01  .3247E+01  .2618E+03
TD450   .2991E+02  .2377E+02  .3987E+01  .2994E+02  .2415E+02  .1912E+01  .3219E+01  .2456E+03
TD500   .2722E+02  .2139E+02  .4067E+01  .3213E+02  .2206E+02  .1903E+01  .3192E+01  .2305E+03
PH0     .3996E+01  .2740E-01 '.4208E+00  .3167E+01  .3998E+01  .1007E+01  .3941E+01  .4106E+01
PH50    .4158E+01  .1157E-01  .1712E+01  .6257E+01  .4158E+01  .1003E+01  .4143E+01  .4212E+01
PH100   .4562E+01  .4458E+00  .3456E+01  .1477E+02  .4544E+01  .1089E+01  .4282E+01  .6878E+01
PH150   .6325E+01  .8984E+00 -.9121E+00  .2255E+01  .6254E+01  .1167E+01  .4419E+01  .7217E+01
PH200   .7082E+01  .2521E+00 -.5366E+01  .4223E+02  .7077E+01  .1041E+01  .4695E+01  .7260E+01
PH250   .7209E+01  .6984E-01 -.3915E+01  .3071E+02  .7207E+01  .1010E+01  .6486E+01  .7268E+01
PH300   .7245E+01  .3231E-01 -.3567E+01  .2377E+02  .7245E+01  .1005E+01  .6945E+01  .7269E+01
PH350   .7259E+01  .1663E-01 -.4709E+01  .3713E+02  .7259E+01  .1002E+01  .7082E+01  .7269E+01
PH400   .7265E+01  .8663E-02 -.3926E+01  .5484E+02  .7265E+01  .1001E+01  .7161E+01  .7269E+01
PH450   .7267E+01  .4669E-02 -.7275E+01  .7791E+02  .7267E+01  .1001E+01  .7205E+01  .7269E+01
PH500   .7268E+01  .2592E-02 -.8712E+01  .1062E+03  .7268E+01  .1000E+01  .7231E+01  .7269E+01
SULPRD  .1509E-05  .6117E-06  .9323E+00  .3877E+01  .1395E-05  .1493E+01  .4623E-06  .3924E-05
RHOTAL  .1392E+07  .1024E+06  .5550E-02  .2642E+01  .1388E+07  .1077E+01  .1129E+07  .1660E+07
O2DIFF  .7573E+02  .2156E+02  .2489E+00  .2330E+01  .7259E+02  .1346E+01  .3152E+02  .1290E+03
KPRIME  .5836E-01  .3066E-01  .1720E+01  .7162E+01  .5189E-01  .1610E+01  .2007E-01  .2261E+00
NPRCP   .3998E+00  .3934E-02 -.5323E-01  .2652E+01  .3998E+00  .1010E+01  .3900E+00  .4099E+00
FACTCO  .3310E+00  .6170E-01  .3415E+00  .2497E+01  .3253E+00  .1205E+01  .2059E+00  .4971E+00
        .5276E-1   .2607E-1   .1278      .1831      .4520E-1   .1822E1    .1003E-1   .9998E-1
```

```
SUBDR   .1535E-03  .2178E-03  .3625E+01  .2161E+02  .7702E-04  .3353E+01  .2201E-05  .2067E-02
FVOIG   .3990E+00  .2002E-01 -.6719E-01  .2406E+01  .3985E+00  .1052E+01  .3550E+00  .4459E+00
RADGP   .5000E-01  .0000E+00  .0000E+00  .0000E+00  .5000E-01  .1000E+01  .5000E-01  .5000E-01
PYRITE  .7300E+03  .0000E+00  .0000E+00  .0000E+00  .7300E+03  .1000E+01  .7300E+03  .7300E+03
UPERTH  .3000E+01  .0000E+00  .0000E+00  .0000E+00  .3000E+01  .1000E+01  .3000E+01  .3000E+01
02TGN   .1200E+01  .0000E+00  .0000E+00  .0000E+00  .1200E+01  .1000E+01  .1200E+01  .1200E+01
RAINTH  .9000E+00  .2029E-01  .5141E-01  .2471E+01  .8995E+00  .1023E+01  .8516E+00  .9466E+00
HDISS   .2007E-01  .5719E-02 -.1964E-01  .1777E+01  .1926E-01  .1336E+01  .1001E-01  .2986E-01
RAINGP  .2744E+00  .1135E+00  .5223E+00  .2532E+01  .2507E+00  .1548E+01  .7316E-01  .5821E+00
CONCRA  .1035E+02  .2099E+01 -.2356E+00  .2306E+01  .1012E+02  .1242E+01  .5434E+01  .1496E+02
HFRAC   .2993E+00  .3975E-01 -.5944E-01  .2417E+01  .2966E+00  .1145E+01  .2031E+00  .3934E+00
TFINAL  .5009E+00  .2863E+00  .1056E+01  .3540E+01  .4285E+00  .1757E+01  .1300E+00  .1397E+01
K3      .2194E-01  .3618E-02  .4704E+00  .2732E+01  .2165E-01  .1177E+01  .1541E-01  .3163E-01
BETA    .1879E+00  .5042E-01  .5423E+00  .2695E+01  .1813E+00  .1306E+01  .1004E+00  .3248E+00
UF      .3469E-04  .4876E-04  .3923E+01  .2618E+02  .1742E-04  .3369E+01  .1118E-05  .4657E-03
PCH20   .1492E+02  .4103E+01 -.3276E-01  .2383E+01  .1431E+02  .1351E+01  .5542E+01  .2486E+02
EMANAT  .3819E+00  .1946E+00  .3855E+00  .2274E+01  .3264E+00  .1829E+01  .3627E-01  .8665E+00
PREC    .1230E+03  .7430E+00 -.2761E-01  .2544E+01  .1230E+03  .1006E+01  .1213E+03  .1249E+03
FWIND1  .3501E+02  .2980E+00 -.8481E-01  .2591E+01  .3501E+02  .1009E+01  .3417E+02  .3579E+02
WIND2   .1000E+00  .7788E-03  .6454E-01  .2842E+01  .1000E+00  .1008E+01  .9915E-01  .1021E+00
GRC     .1978E+01  .5752E+00  .3156E-01  .1802E+01  .1890E+01  .1362E+01  .1006E+01  .2996E+01
UWIND   .5300E+01  .2153E-01  .2639E+00  .3238E+01  .5300E+01  .1004E+01  .5246E+01  .5367E+01
DISP    .5988E+02  .4466E+01  .1244E+00  .2799E+01  .5972E+02  .1078E+01  .4921E+02  .7300E+02
DEPVEL  .1629E-01  .7572E-02  .3553E-01  .1851E+01  .1419E-01  .1764E+01  .3029E-02  .2998E-01
DEPSF1  .8939E+00  .5770E-01  .8730E-01  .1812E+01  .8921E+00  .1067E+01  .8002E+00  .9994E+00
DEPSF2  .4941E+00  .2476E+00  .3751E-02  .2160E+01  .4014E+00  .2194E+01  .1005E-01  .9982E+00
FDEPSF3 .2009E+00  .5744E-01  .8095E-01  .1844E+01  .1923E+00  .1351E+01  .1007E+00  .2999E+00
EPSF4   .1998E+00  .5979E-01  .1580E-01  .1701E+01  .1904E+00  .1375E+01  .1001E+00  .3000E+00
FDEPED1 .8961E+00  .5661E-01  .1926E-01  .1384E+01  .8943E+00  .1065E+01  .5000E+00  .9993E+00
EFED2   .9026E+00  .5683E-01 -.4947E-01  .1841E+01  .9008E+00  .1065E+01  .8001E+00  .9996E+00
EPED3   .9009E+00  .5771E-01 -.8673E-01  .1840E+01  .8990E+00  .1067E+01  .8003E+00  .9998E+00
FDEPED4 .9950E-01  .2892E-01  .3708E-01  .1813E+01  .9508E-01  .1362E+01  .5024E-01  .1499E+00
```

```
FDEPED4  .9950E-01  .2892E-01  .3708E-01  .1813E+01  .9508E-01  .1352E+01  .5024E-01  .1499E+00

DUREXP1  .9997E+08  .5762E+07 -.3851E-01  .1753E+01  .9981E+08  .1060E+01  .9003E+08  .1099E+09

DUREXP2  .2900E+07  .1142E+06 -.3262E-01  .1822E+01  .2898E+07  .1040E+01  .2701E+07  .3099E+07

DUREXP3  .5146E+07  .2745E+06 -.2839E-01  .1667E+01  .5139E+07  .1055E+01  .4701E+07  .5600E+07

DUREXP4  .5140E+07  .2623E+06  .1855E-01  .1838E+01  .5133E+07  .1052E+01  .4705E+07  .5600E+07

VEGLAM1  .3023E-08  .5932E-09 -.8152E-01  .1805E+01  .2962E-08  .1226E+01  .2001E-08  .3997E-08

VEGLAM2  .5004E-06  .5682E-07  .4192E-02  .1814E+01  .4972E-06  .1121E+01  .4002E-06  .5998E-06

VEGLAM3  .4986E-06  .5969E-07  .8578E-01  .1703E+01  .4950E-06  .1128E+01  .4002E-06  .6000E-06

VEGLAM4  .4989E-06  .5636E-07  .1154E+00  .1857E+01  .4957E-06  .1120E+01  .4011E-06  .5996E-06

YDEN1    .5000E+03  .5630E+02 -.7012E-01  .1846E+01  .4968E+03  .1121E+01  .4001E+03  .5990E+03

YDEN2    .7467E+03  .1476E+03  .7992E-01  .1783E+01  .7320E+03  .1223E+01  .5019E+03  .9994E+03

YDEN3    .1994E+04  .2879E+03  .4176E-02  .1807E+01  .1973E+04  .1158E+01  .1501E+04  .2500E+04

YDEN4    .2002E+04  .2931E+03 -.2207E-01  .1783E+01  .1980E+04  .1161E+01  .1500E+04  .2499E+04
 FBI
~~FAUT1~~    .1411E-08  .6505E-09  .4242E-03  .1782E+01  .1232E-08  .1755E+01  .3027E-09  .2500E-08

FAUT1    .2497E+00  .9318E-01  .2626E+00  .2398E+01  .2308E+00  .1516E+01  .5690E-01  .4856E+00

FAUT2    .2529E+00  .8940E-01  .3155E+00  .2533E+01  .2361E+00  .1469E+01  .6858E-01  .4870E+00

FACTA    .7526E+00  .1427E+00 -.6034E-01  .1887E+01  .7387E+00  .1217E+01  .5011E+00  .9992E+00

FACTB    .7464E+00  .1466E+00  .4530E-01  .1773E+01  .7317E+00  .1223E+01  .5005E+00  .9985E+00

FLOCA    .3368E+00  .9770E-01  .1684E+00  .2564E+01  .3217E+00  .1367E+01  .1198E+00  .5828E+00

FMEAT    .5003E+00  .2833E+00  .1236E-01  .1876E+01  .3730E+00  .2680E+01  .1249E-02  .9996E+00

FISHIN   .8032E+04  .2383E+04  .3690E-01  .1720E+01  .7660E+04  .1370E+01  .4010E+04  .1196E+05

DF       .6401E+00  .1054E+00  .1733E+00  .2319E+01  .6314E+00  .1181E+01  .4074E+00  .8876E+00

F        .4244E+00  .1007E+00  .3285E+00  .2584E+01  .4124E+00  .1274E+01  .2115E+00  .6915E+00

B:'B
```

TABLE 3

```
YPE. NOMINAL2.DAT
ROBABILISTIC ANALYSIS OF SOIL CONC
  0  97   0   0   0
0913566091    1
WIND1     1       3.5E1
FWIND2    1       .1E00
FPREC     1      1.23E2
WIND      1      5.3E00
YEARS     1       5.0E2
DT        1      1.0E01
TIME      1      5.0E01
DDS       1      0.0E00
CODL      1      0.0E00
CODPH     1      0.0E00
DDF       1      0.0E00
CODR      1      0.0E00
CODD      1      0.0E00
DDA       1      0.0E00
DDV1      1      0.0E00
CODV2     1      0.0E00
DDO       1      0.0E00
DDU       1      0.0E00
SULPRD    1      1.4E-6
RHOTAL    1       1.4E6
2CONC     1        9.04
2DIFF     1        70.0
KPRIME    1      5.0E-2
SUBSO     1       2.5E1
PRCP      1      4.0E-1
FACTCO    1      3.0E-1
DISTCO    1      5.0E-2
UNSAT     1         1.0
DMIX      1         1.0
KSUBDR    1     80.0E-6
AINGP     1      2.0E-1
VOID      1      4.0E-1
CONCRA    1        11.0
RACGP     1      5.0E-2
YRITE     1       7.3E2
UPERTH    1      3.0E00
O2TON     1      1.2E00
AINTH     1      9.0E-1
THDISS    1      2.0E-2
THFRAC    1         0.3
K1        1      4.3E-1
K3        1      2.2E-2
BETA      1      1.8E-1
LAKE      1      1.6E06
LAKE      1      1.0E01
AWAT      1      2.0E07
ATAIL     1      5.0E05
CTCO      1      0.8E00
BF        1     1.60E-5
MPCH2O    1      1.5E01
F LAM     1      2.1E-6
L MANAT   1    3.25E-1
SILT      1        .3E2
EXPT      1       5.0E5
H RC      1      2.0E00
XDIST     1      3.6E03
DISP      1      6.0E01
F EPSF1   1      9.0E-1
FDEPSF2   1      4.0E-1
FDEPSF3   1      2.0E-1
```

```
 FDEPSF4   1       2.0E-1
 FDEPED1   1       9.0E-1
 FDEPED2   1       9.0E-1
 FDEPED3   1       9.0E-1
 FDEPED4   1       1.0E-1
 DUREXP1   1       1.0E08
 DUREXP2   1       2.9E06
 DUREXP3   1      5.15E06
 DUREXP4   1      5.15E06
 DEPVEL    1       1.4E-2
 VEGLAM1   1       3.0E-9
 VEGLAM2   1       5.0E-7
 VEGLAM3   1       5.0E-7
 VEGLAM4   1       5.0E-7
 YDEN1     1       5.0E02
 YDEN2     1       7.5E02
 YDEN3     1       2.0E03
 YDEN4     1       2.0E03
 FBI       1       1.2E-9
 FAUT1     1       .23E00
 FAUT2     1       .23E00
 FIR       1      2.555E3
 DCFING    1    3.200E-7
 VEGINA    1    1.570E05
 VEGINB    1       7.0E04
 FACTA     1       7.5E-1
 FACTB     1       7.5E-1
 FLOCA     1       .32E00
 XMEATI    1       7.9E04
 FMEAT     1       0.5E00
 WATIN     1        .6E00
 FISHIN    1       8.0E03
 FFISH     1       1.0E00
 DCFINH    1       2.6E-6
 XINHIN    1       8.4E03
 OF        1       6.0E-1
 F         1       4.0E-1

B:\B
```

```
:\RISKC
:\RISKC
```

MAR 16
Nominal Values.

TABLE 4.

```
PAC CALLED ON TRIAL NUMBER    1

SUMMARY RESULTS

 RIABLE    MEAN      STD DEV     SKEWNESS    KURTOSIS   GEOM MEAN    GEOM SD     MINIMUM    MAXIMUM

DV0     .2432E+01  .0000E+00  .0000E+00  .0000E+00  .2432E+01  .1000E+01  .2432E+01  .2432E+01

  50    .2397E+01  .0000E+00  .0000E+00  .0000E+00  .2397E+01  .1000E+01  .2397E+01  .2397E+01

DV100   .2364E+01  .0000E+00  .0000E+00  .0000E+00  .2364E+01  .1000E+01  .2364E+01  .2364E+01

  150   .2330E+01  .0000E+00  .0000E+00  .0000E+00  .2330E+01  .1000E+01  .2330E+01  .2330E+01

  200   .2298E+01  .0000E+00  .0000E+00  .0000E+00  .2298E+01  .1000E+01  .2298E+01  .2298E+01

DV250   .2266E+01  .0000E+00  .0000E+00  .0000E+00  .2266E+01  .1000E+01  .2266E+01  .2266E+01

  300   .2235E+01  .0000E+00  .0000E+00  .0000E+00  .2235E+01  .1000E+01  .2235E+01  .2235E+01

DV350   .2205E+01  .0000E+00  .0000E+00  .0000E+00  .2205E+01  .1000E+01  .2205E+01  .2205E+01

  400   .2175E+01  .0000E+00  .0000E+00  .0000E+00  .2175E+01  .1000E+01  .2175E+01  .2175E+01

  450   .2146E+01  .0000E+00  .0000E+00  .0000E+00  .2146E+01  .1000E+01  .2146E+01  .2146E+01

DV500   .2118E+01  .0000E+00  .0000E+00  .0000E+00  .2118E+01  .1000E+01  .2118E+01  .2118E+01
```

```
DM50     .1429E-02  .0000E+00  .0000E+00  .0000E+00  .1429E-02  .1000E+01  .1429E-02  .1429E-02

DM100    .1408E-02  .0000E+00  .0000E+00  .0000E+00  .1408E-02  .1000E+01  .1408E-02  .1408E-02

DM150    .1389E-02  .0000E+00  .0000E+00  .0000E+00  .1389E-02  .1000E+01  .1389E-02  .1389E-02

DM200    .1370E-02  .0000E+00  .0000E+00  .0000E+00  .1370E-02  .1000E+01  .1370E-02  .1370E-02

DM250    .1351E-02  .0000E+00  .0000E+00  .0000E+00  .1351E-02  .1000E+01  .1351E-02  .1351E-02

DM300    .1332E-02  .0000E+00  .0000E+00  .0000E+00  .1332E-02  .1000E+01  .1332E-02  .1332E-02

DM350    .1314E-02  .0000E+00  .0000E+00  .0000E+00  .1314E-02  .1000E+01  .1314E-02  .1314E-02

DM400    .1296E-02  .0000E+00  .0000E+00  .0000E+00  .1296E-02  .1000E+01  .1296E-02  .1296E-02

DM450    .1279E-02  .0000E+00  .0000E+00  .0000E+00  .1279E-02  .1000E+01  .1279E-02  .1279E-02

DM500    .1262E-02  .0000E+00  .0000E+00  .0000E+00  .1262E-02  .1000E+01  .1262E-02  .1262E-02

DW0      .4931E+01  .0000E+00  .0000E+00  .0000E+00  .4931E+01  .1000E+01  .4931E+01  .4931E+01

DW50     .6751E+01  .0000E+00  .0000E+00  .0000E+00  .6751E+01  .1000E+01  .6751E+01  .6751E+01

DW100    .1039E+02  .0000E+00  .0000E+00  .0000E+00  .1039E+02  .1000E+01  .1039E+02  .1039E+02

DW150    .3237E+02  .0000E+00  .0000E+00  .0000E+00  .3237E+02  .1000E+01  .3237E+02  .3237E+02

DW200    .2898E+02  .0000E+00  .0000E+00  .0000E+00  .2898E+02  .1000E+01  .2898E+02  .2898E+02

DW250    .2454E+02  .0000E+00  .0000E+00  .0000E+00  .2454E+02  .1000E+01  .2454E+02  .2454E+02

DW300    .2074E+02  .0000E+00  .0000E+00  .0000E+00  .2074E+02  .1000E+01  .2074E+02  .2074E+02

DW350    .1756E+02  .0000E+00  .0000E+00  .0000E+00  .1756E+02  .1000E+01  .1756E+02  .1756E+02

DW400    .1489E+02  .0000E+00  .0000E+00  .0000E+00  .1489E+02  .1000E+01  .1489E+02  .1489E+02

DW450    .1266E+02  .0000E+00  .0000E+00  .0000E+00  .1266E+02  .1000E+01  .1266E+02  .1266E+02

DW500    .1079E+02  .0000E+00  .0000E+00  .0000E+00  .1079E+02  .1000E+01  .1079E+02  .1079E+02

DF0      .1753E+01  .0000E+00  .0000E+00  .0000E+00  .1753E+01  .1000E+01  .1753E+01  .1753E+01

DF50     .2400E+01  .0000E+00  .0000E+00  .0000E+00  .2400E+01  .1000E+01  .2400E+01  .2400E+01

DF100    .3693E+01  .0000E+00  .0000E+00  .0000E+00  .3693E+01  .1000E+01  .3693E+01  .3693E+01

DF150    .1151E+02  .0000E+00  .0000E+00  .0000E+00  .1151E+02  .1000E+01  .1151E+02  .1151E+02

DF200    .1031E+02  .0000E+00  .0000E+00  .0000E+00  .1031E+02  .1000E+01  .1031E+02  .1031E+02

DF250    .8724E+01  .0000E+00  .0000E+00  .0000E+00  .8724E+01  .1000E+01  .8724E+01  .8724E+01

DF300    .7374E+01  .0000E+00  .0000E+00  .0000E+00  .7374E+01  .1000E+01  .7374E+01  .7374E+01

DF350    .6242E+01  .0000E+00  .0000E+00  .0000E+00  .6242E+01  .1000E+01  .6242E+01  .6242E+01

DF400    .5295E+01  .0000E+00  .0000E+00  .0000E+00  .5295E+01  .1000E+01  .5295E+01  .5295E+01

DF450    .4502E+01  .0000E+00  .0000E+00  .0000E+00  .4502E+01  .1000E+01  .4502E+01  .4502E+01

DF500    .3838E+01  .0000E+00  .0000E+00  .0000E+00  .3838E+01  .1000E+01  .3838E+01  .3838E+01
```

```
D[illegible]0    .1066E+01  .0000E+00  .0000E+00  .0000E+00  .1066E+01  .1000E+01  .1066E+01  .1066E+01
DD100   .1051E+01  .0000E+00  .0000E+00  .0000E+00  .1051E+01  .1000E+01  .1051E+01  .1051E+01
D[illegible]50   .1037E+01  .0000E+00  .0000E+00  .0000E+00  .1037E+01  .1000E+01  .1037E+01  .1037E+01
DD200   .1022E+01  .0000E+00  .0000E+00  .0000E+00  .1022E+01  .1000E+01  .1022E+01  .1022E+01
D[illegible]50   .1008E+01  .0000E+00  .0000E+00  .0000E+00  .1008E+01  .1000E+01  .1008E+01  .1008E+01
D[illegible]00   .9943E+00  .0000E+00  .0000E+00  .0000E+00  .9943E+00  .1000E+01  .9943E+00  .9943E+00
DD350   .9808E+00  .0000E+00  .0000E+00  .0000E+00  .9808E+00  .1000E+01  .9808E+00  .9808E+00
D[illegible]00   .9676E+00  .0000E+00  .0000E+00  .0000E+00  .9676E+00  .1000E+01  .9676E+00  .9676E+00
DD450   .9546E+00  .0000E+00  .0000E+00  .0000E+00  .9546E+00  .1000E+01  .9546E+00  .9546E+00
D[illegible]00   .9420E+00  .0000E+00  .0000E+00  .0000E+00  .9420E+00  .1000E+01  .9420E+00  .9420E+00
D[illegible]     .7711E+01  .0000E+00  .0000E+00  .0000E+00  .7711E+01  .1000E+01  .7711E+01  .7711E+01
DR50    .6807E+01  .0000E+00  .0000E+00  .0000E+00  .6807E+01  .1000E+01  .6807E+01  .6807E+01
D[illegible]00   .6425E+01  .0000E+00  .0000E+00  .0000E+00  .6425E+01  .1000E+01  .6425E+01  .6425E+01
DR150   .5912E+01  .0000E+00  .0000E+00  .0000E+00  .5912E+01  .1000E+01  .5912E+01  .5912E+01
D[illegible]200  .5829E+01  .0000E+00  .0000E+00  .0000E+00  .5829E+01  .1000E+01  .5829E+01  .5829E+01
D[illegible]250  .5749E+01  .0000E+00  .0000E+00  .0000E+00  .5749E+01  .1000E+01  .5749E+01  .5749E+01
DR300   .5670E+01  .0000E+00  .0000E+00  .0000E+00  .5670E+01  .1000E+01  .5670E+01  .5670E+01
D[illegible]50   .5594E+01  .0000E+00  .0000E+00  .0000E+00  .5594E+01  .1000E+01  .5594E+01  .5594E+01
DR400   .5518E+01  .0000E+00  .0000E+00  .0000E+00  .5518E+01  .1000E+01  .5518E+01  .5518E+01
D[illegible]50   .5444E+01  .0000E+00  .0000E+00  .0000E+00  .5444E+01  .1000E+01  .5444E+01  .5444E+01
D[illegible]500  .5372E+01  .0000E+00  .0000E+00  .0000E+00  .5372E+01  .1000E+01  .5372E+01  .5372E+01
TD0     .1791E+02  .0000E+00  .0000E+00  .0000E+00  .1791E+02  .1000E+01  .1791E+02  .1791E+02
T[illegible]0    .1942E+02  .0000E+00  .0000E+00  .0000E+00  .1942E+02  .1000E+01  .1942E+02  .1942E+02
TD100   .2392E+02  .0000E+00  .0000E+00  .0000E+00  .2392E+02  .1000E+01  .2392E+02  .2392E+02
T[illegible]50   .5315E+02  .0000E+00  .0000E+00  .0000E+00  .5315E+02  .1000E+01  .5315E+02  .5315E+02
T[illegible]200  .4844E+02  .0000E+00  .0000E+00  .0000E+00  .4844E+02  .1000E+01  .4844E+02  .4844E+02
TD250   .4229E+02  .0000E+00  .0000E+00  .0000E+00  .4229E+02  .1000E+01  .4229E+02  .4229E+02
T[illegible]00   .3701E+02  .0000E+00  .0000E+00  .0000E+00  .3701E+02  .1000E+01  .3701E+02  .3701E+02
TD350   .3258E+02  .0000E+00  .0000E+00  .0000E+00  .3258E+02  .1000E+01  .3258E+02  .3258E+02
T[illegible]00   .2885E+02  .0000E+00  .0000E+00  .0000E+00  .2885E+02  .1000E+01  .2885E+02  .2885E+02
T[illegible]50   .2571E+02  .0000E+00  .0000E+00  .0000E+00  .2571E+02  .1000E+01  .2571E+02  .2571E+02
TD500   .2307E+02  .0000E+00  .0000E+00  .0000E+00  .2307E+02  .1000E+01  .2307E+02  .2307E+02
```

```
PH50    .4153E+01  .0000E+00  .0000E+00  .0000E+00  .4153E+01  .1000E+01  .4153E+01  .4153E+01
PH100   .4423E+01  .0000E+00  .0000E+00  .0000E+00  .4423E+01  .1000E+01  .4423E+01  .4423E+01
PH150   .6747E+01  .0000E+00  .0000E+00  .0000E+00  .6747E+01  .1000E+01  .6747E+01  .6747E+01
PH200   .7150E+01  .0000E+00  .0000E+00  .0000E+00  .7150E+01  .1000E+01  .7150E+01  .7150E+01
PH250   .7227E+01  .0000E+00  .0000E+00  .0000E+00  .7227E+01  .1000E+01  .7227E+01  .7227E+01
PH300   .7256E+01  .0000E+00  .0000E+00  .0000E+00  .7256E+01  .1000E+01  .7256E+01  .7256E+01
PH350   .7265E+01  .0000E+00  .0000E+00  .0000E+00  .7265E+01  .1000E+01  .7265E+01  .7265E+01
PH400   .7268E+01  .0000E+00  .0000E+00  .0000E+00  .7268E+01  .1000E+01  .7268E+01  .7268E+01
PH450   .7269E+01  .0000E+00  .0000E+00  .0000E+00  .7269E+01  .1000E+01  .7269E+01  .7269E+01
PH500   .7269E+01  .0000E+00  .0000E+00  .0000E+00  .7269E+01  .1000E+01  .7269E+01  .7269E+01
SULPRD  .1400E-05  .0000E+00  .0000E+00  .0000E+00  .1400E-05  .1000E+01  .1400E-05  .1400E-05
RHOTAL  .1400E+07  .0000E+00  .0000E+00  .0000E+00  .1400E+07  .1000E+01  .1400E+07  .1400E+07
O2DIFF  .7000E+02  .0000E+00  .0000E+00  .0000E+00  .7000E+02  .1000E+01  .7000E+02  .7000E+02
KPRIME  .5000E-01  .0000E+00  .0000E+00  .0000E+00  .5000E-01  .1000E+01  .5000E-01  .5000E-01
NPRCP   .4000E+00  .0000E+00  .0000E+00  .0000E+00  .4000E+00  .1000E+01  .4000E+00  .4000E+00
FACTCO  .3000E+00  .0000E+00  .0000E+00  .0000E+00  .3000E+00  .1000E+01  .3000E+00  .3000E+00
DISTCO  .5000E-01  .0000E+00  .0000E+00  .0000E+00  .5000E-01  .1000E+01  .5000E-01  .5000E-01
DUNSAT  .1000E+01  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .1000E+01  .1000E+01
KSUBDR  .8000E-04  .0000E+00  .0000E+00  .0000E+00  .8000E-04  .1000E+01  .8000E-04  .8000E-04
TVOID   .4000E+00  .0000E+00  .0000E+00  .0000E+00  .4000E+00  .1000E+01  .4000E+00  .4000E+00
FRACGP  .5000E-01  .0000E+00  .0000E+00  .0000E+00  .5000E-01  .1000E+01  .5000E-01  .5000E-01
PYRITE  .7300E+03  .0000E+00  .0000E+00  .0000E+00  .7300E+03  .1000E+01  .7300E+03  .7300E+03
UPERTH  .3000E+01  .0000E+00  .0000E+00  .0000E+00  .3000E+01  .1000E+01  .3000E+01  .3000E+01
UO2TON  .1200E+01  .0000E+00  .0000E+00  .0000E+00  .1200E+01  .1000E+01  .1200E+01  .1200E+01
RAINTH  .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00
THDISS  .2000E-01  .0000E+00  .0000E+00  .0000E+00  .2000E-01  .1000E+01  .2000E-01  .2000E-01
RAINGP  .2000E+00  .0000E+00  .0000E+00  .0000E+00  .2000E+00  .1000E+01  .2000E+00  .2000E+00
CONCRA  .1100E+02  .0000E+00  .0000E+00  .0000E+00  .1100E+02  .1000E+01  .1100E+02  .1100E+02
THFRAC  .3000E+00  .0000E+00  .0000E+00  .0000E+00  .3000E+00  .1000E+01  .3000E+00  .3000E+00
TFINAL  .4300E+00  .0000E+00  .0000E+00  .0000E+00  .4300E+00  .1000E+01  .4300E+00  .4300E+00
XK3     .2200E-01  .0000E+00  .0000E+00  .0000E+00  .2200E-01  .1000E+01  .2200E-01  .2200E-01
BETA    .1800E+00  .0000E+00  .0000E+00  .0000E+00  .1800E+00  .1000E+01  .1800E+00  .1800E+00
```

```
BF      .1600E-04  .0000E+00  .0000E+00  .0000E+00  .1600E-04  .1000E+01  .1600E-04  .1600E-04

[redacted]CH2O   .1500E+02  .0000E+00  .0000E+00  .0000E+00  .1500E+02  .1000E+01  .1500E+02  .1500E+02

EMANAT  .3250E+00  .0000E+00  .0000E+00  .0000E+00  .3250E+00  .1000E+01  .3250E+00  .3250E+00

[redacted]REC    .1230E+03  .0000E+00  .0000E+00  .0000E+00  .1230E+03  .1000E+01  .1230E+03  .1230E+03

[redacted]WIND1  .3500E+02  .0000E+00  .0000E+00  .0000E+00  .3500E+02  .1000E+01  .3500E+02  .3500E+02

FWIND2  .1000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+00  .1000E+01  .1000E+00  .1000E+00

[redacted]RC     .2000E+01  .0000E+00  .0000E+00  .0000E+00  .2000E+01  .1000E+01  .2000E+01  .2000E+01

UWIND   .5300E+01  .0000E+00  .0000E+00  .0000E+00  .5300E+01  .1000E+01  .5300E+01  .5300E+01

[redacted]ISP    .6000E+02  .0000E+00  .0000E+00  .0000E+00  .6000E+02  .1000E+01  .6000E+02  .6000E+02

[redacted]EPVEL  .1400E-01  .0000E+00  .0000E+00  .0000E+00  .1400E-01  .1000E+01  .1400E-01  .1400E-01

[redacted]DEPSF  .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00

[redacted]EPSF   .4000E+00  .0000E+00  .0000E+00  .0000E+00  .4000E+00  .1000E+01  .4000E+00  .4000E+00

FDEPSF  .2000E+00  .0000E+00  .0000E+00  .0000E+00  .2000E+00  .1000E+01  .2000E+00  .2000E+00

[redacted]EPSF   .2000E+00  .0000E+00  .0000E+00  .0000E+00  .2000E+00  .1000E+01  .2000E+00  .2000E+00

[redacted]DEPED  .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00

[redacted]DEPED  .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00

[redacted]EPED   .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00

FDEPED  .1000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+00  .1000E+01  .1000E+00  .1000E+00

[redacted]REXP   .1000E+09  .0000E+00  .0000E+00  .0000E+00  .1000E+09  .1000E+01  .1000E+09  .1000E+09

[redacted]UREXP  .2900E+07  .0000E+00  .0000E+00  .0000E+00  .2900E+07  .1000E+01  .2900E+07  .2900E+07

[redacted]UREXP  .5150E+07  .0000E+00  .0000E+00  .0000E+00  .5150E+07  .1000E+01  .5150E+07  .5150E+07

[redacted]REXP   .5150E+07  .0000E+00  .0000E+00  .0000E+00  .5150E+07  .1000E+01  .5150E+07  .5150E+07

VEGLAM  .3000E-08  .0000E+00  .0000E+00  .0000E+00  .3000E-08  .1000E+01  .3000E-08  .3000E-08

[redacted]GLAM   .5000E-06  .0000E+00  .0000E+00  .0000E+00  .5000E-06  .1000E+01  .5000E-06  .5000E-06

[redacted]EGLAM  .5000E-06  .0000E+00  .0000E+00  .0000E+00  .5000E-06  .1000E+01  .5000E-06  .5000E-06

[redacted]EGLAM  .5000E-06  .0000E+00  .0000E+00  .0000E+00  .5000E-06  .1000E+01  .5000E-06  .5000E-06

[redacted]EN1    .5000E+03  .0000E+00  .0000E+00  .0000E+00  .5000E+03  .1000E+01  .5000E+03  .5000E+03

YDEN2   .7500E+03  .0000E+00  .0000E+00  .0000E+00  .7500E+03  .1000E+01  .7500E+03  .7500E+03

[redacted]EN3    .2000E+04  .0000E+00  .0000E+00  .0000E+00  .2000E+04  .1000E+01  .2000E+04  .2000E+04

[redacted]DEN4   .2000E+04  .0000E+00  .0000E+00  .0000E+00  .2000E+04  .1000E+01  .2000E+04  .2000E+04

[redacted]BI     .1200E-08  .0000E+00  .0000E+00  .0000E+00  .1200E-08  .1000E+01  .1200E-08  .1200E-08

[redacted]UT1    .2300E+00  .0000E+00  .0000E+00  .0000E+00  .2300E+00  .1000E+01  .2300E+00  .2300E+00

FAUT2   .2300E+00  .0000E+00  .0000E+00  .0000E+00  .2300E+00  .1000E+01  .2300E+00  .2300E+00
```

```
FACTB   .7500E+00  .0000E+00  .0000E+00  .0000E+00  .7500E+00  .1000E+01  .7500E+00  .7500E+00

FLOCA   .3200E+00  .0000E+00  .0000E+00  .0000E+00  .3200E+00  .1000E+01  .3200E+00  .3200E+00

FMEAT   .5000E+00  .0000E+00  .0000E+00  .0000E+00  .5000E+00  .1000E+01  .5000E+00  .5000E+00

FISHIN  .8000E+04  .0000E+00  .0000E+00  .0000E+00  .8000E+04  .1000E+01  .8000E+04  .8000E+04

OF      .6000E+00  .0000E+00  .0000E+00  .0000E+00  .6000E+00  .1000E+01  .6000E+00  .6000E+00

F       .4000E+00  .0000E+00  .0000E+00  .0000E+00  .4000E+00  .1000E+01  .4000E+00  .4000E+00
Stop - Program terminated.

B:\B
```

```
FDEPSF                                               :
FDEPSF                                               :
FDEPSF                                               :
FDEPSF                                               :
FDEPED                                               :
FDEPED                                               :
FDEPED                                               :
FDEPED                                               :
DUREXP                                               :
DUREXP                                               :
DUREXP                                               :
DUREXP                                               :
VEGLAM                                               :
VEGLAM                                               :
VEGLAM                                               :
VEGLAM                                               :
FDEN1                                                :
FDEN2                                                :
FDEN3                                                :
FDEN4                                                :
FOUT1                                                :
FOUT2                                                :
F  T                                                 :
FACTA                                                :
FACTR                                                :
FLOCA                                                :
FHEAT                                                :
FISHIN                                               :
F                                                    :
F                                                    :

  NSTART                   NIN           NOUT          NPARM         NITER
 913566071                   0            150            97              1
```

```
DR350   DOSE AT T=350 RADON INHALATION          :
DR400   DOSE AT T=400 RADON INHALATION          :
DR450   DOSE AT T=450 RADON INHALATION          :
DR500   DOSE AT T=500 RADON INHALATION          :
TD0     TOTAL DOSE AT T=0                       :
TD50    TOTAL DOSE AT T = 50                    :
TD100   TOTAL DOSE AT T=100                     :
TD150   TOTAL DOSE AT T=150                     :
TD200   TOTAL DOSE AT T=200                     :
TD250   TOTAL DOSE AT T=250                     :
TD300   TOTAL DOSE AT T=300                     :
TD350   TOTAL DOSE AT T=350                     :
TD400   TOTAL DOSE AT T=400                     :
TD450   TOTAL DOSE AT T=450                     :
TD500   TOTAL DOSE AT T=500                     :
PH0     PH AT T=0                               :
PH50    PH AT T=50                              :
PH100   PH AT T=100                             :
PH150   PH AT T=150                             :
PH200   PH AT T=200                             :
PH250   PH AT T=250                             :
PH300   PH AT T=300                             :
PH350   PH AT T=350                             :
PH400   PH AT T=400                             :
PH450   PH AT T=450                             :
PH500   PH AT T=500                             :
SULPRD                                          :
RHOTAL                                          :
O2DIFF                                          :
KPRIME                                          :
NFRCP                                           :
FACTCO                                          :
DISTCO                                          :
OUNSAT                                          :
KSURDR                                          :
TVOID                                           :
FRACGP                                          :
PYRITE                                          :
UPERTH                                          :
UO2TON                                          :
RATNTH                                          :
THDISS                                          :
RATNGP                                          :
CONCRA                                          :
THFRAC                                          :
TFINAL                                          :
XK3                                             :
BETA                                            :
BF                                              :
NPCH2O                                          :
EMANAT                                          :
FPREC                                           :
FWIND1                                          :
FWIND2                                          :
HSRC                                            :
UWIND                                           :
VDISP                                           :
DEPVEL                                          :
```

TABLE 5.

```
VARIABLE NAMES
*************

DV0      DOSE AT T=0 VEGETATION                       :
DV50     DOSE AT T=50 VEGETATION                      :
DV100    DOSE AT T=100 VEGETATION                     :
DV150    DOSE AT T=150 VEGETATION                     :
DV200    DOSE AT T=200 VEGETATION                     :
DV250    DOSE AT T=250 VEGETATION                     :
DV300    DOSE AT T=300 VEGETATION                     :
DV350    DOSE AT T=350 VEGETATION                     :
DV400    DOSE AT T=400 VEGETATION                     :
DV450    DOSE AT T=450 VEGETATION                     :
DV500    DOSE AT T=500 VEGETATION                     :
DM0      DOSE AT T=0 MEAT                             :
DM50     DOSE AT T=50 MEAT                            :
DM100    DOSE AT T=100 MEAT                           :
DM150    DOSE AT T=150 MEAT                           :
DM200    DOSE AT T=200 MEAT                           :
DM250    DOSE AT T=250 MEAT                           :
DM300    DOSE AT T=300 MEAT                           :
DM350    DOSE AT T=350 MEAT                           :
DM400    DOSE AT T=400 MEAT                           :
DM450    DOSE AT T=450 MEAT                           :
DM500    DOSE AT T=500 MEAT                           :
DW0      DOSE AT T=0 DRINKING WATER                   :
DW50     DOSE AT T=50 DRINKING WATER                  :
DW100    DOSE AT T=100 DRINKING WATER                 :
DW150    DOSE AT T=150 DRINKING WATER                 :
DW200    DOSE AT T=200 DRINKING WATER                 :
DW250    DOSE AT T=250 DRINKING WATER                 :
DW300    DOSE AT T=300 DRINKING WATER                 :
DW350    DOSE AT T=350 DRINKING WATER                 :
DW400    DOSE AT T=400 DRINKING WATER                 :
DW450    DOSE AT T=450 DRINKING WATER                 :
DW500    DOSE AT T=500 DRINKING WATER                 :
DF0      DOSE AT T=0 FISH                             :
DF50     DOSE AT T=50 FISH                            :
DF100    DOSE AT T=100 FISH                           :
DF150    DOSE AT T=150 FISH                           :
DF200    DOSE AT T=200 FISH                           :
DF250    DOSE AT T=250 FISH                           :
DF300    DOSE AT T=300 FISH                           :
DF350    DOSE AT T=350 FISH                           :
DF400    DOSE AT T=400 FISH                           :
DF450    DOSE AT T=450 FISH                           :
DF500    DOSE AT T=500 FISH                           :
DD0      DOSE AT T=0 DUST INHALATION                  :
DD50     DOSE AT T=50 DUST INHALATION                 :
DD100    DOSE AT T=100 DUST INHALATION                :
DD150    DOSE AT T=150 DUST INHALATION                :
DD200    DOSE AT T=200 DUST INHALATION                :
DD250    DOSE AT T=250 DUST INHALATION                :
DD300    DOSE AT T=300 DUST INHALATION                :
DD350    DOSE AT T=350 DUST INHALATION                :
DD400    DOSE AT T=400 DUST INHALATION                :
DD450    DOSE AT T=450 DUST INHALATION                :
DD500    DOSE AT T=500 DUST INHALATION                :
DR0      DOSE AT T=0 RADON INHALATION                 :
DR50     DOSE AT T=50 RADON INHALATION                :
DR100    DOSE AT T=100 RADON INHALATION               :
DR150    DOSE AT T=150 RADON INHALATION               :
DR200    DOSE AT T=200 RADON INHALATION               :
DR250    DOSE AT T=250 RADON INHALATION               :
DR300    DOSE AT T=300 RADON INHALATION               :
```

Call to SENES April 8/86 1:32 p.m.

(Bruce Halbert)

1. Our first attempt at the UTAP-SYVAC3 comparison, for the deterministic run, showed quantitative agreement for XMEATD, TVEGDS, XINDOS and RNDOS. There was qualitative agreement for FISHDS WATDOS, and TDOSE, and obvious disagreement for PHLAKE over most of the time range.

2. We calculate a value for TFINAL of 134a, whereas the value given in table 4 of the SENES deterministic results is 0.43a. Is the SENES value correct? It does not appear to be realistic. BH will check.

3. What are the units for the dose parameters in Table 4? We have assumed that they are μSv/a. BH thinks the units are as assumed, and will confirm.

4. A possible problem in the disagreements observed is a difference in the value of time at the initial condition. We assume, for example, that Ra concentrations in the lake are zero at t=0 (see page C-3), so that WATDOS at t=0 should be zero. SYVAC3 produces a zero dose, but UTAP does not, at t=0. BH explained that the values printed at t=0 correspond to the midpoint of the first step; i.e. either the initial conditions are specified at t=-5a, with output at 0, 50, 100 ...., or the initial conditions are specified at t=0a with output at 5, 55, 105 ... We shall adjust SYVAC3 accordingly, and rerun the deterministic case.

5. The pH differences are affected by the initial conditions, and by a bug recently noted in code for the tailings model in SYVAC3. Another possible problem is in the equations for Ca and SO_4 concentrations in the lake. The equations on page C-6 of Volume 3 - for example should the last term for Ca be:

$$\left\{[Ca]_{L,t}\right\} \exp \left\{\frac{-\Delta t}{T_t}\right\}$$

BH will check the UTAP code to verify whether the equations on page C-6, or some others, have been used.

Return Call from SENES April 9/86 9:30 a.m.
(Bruce Halbert)

1. The units in Table 4 are μSv/a (BH).

2. TFINAL in Table 4 is mislabelled; the quantity given is actually XK1 (BH).

3. The equations on page C-6 are incorrect. The last two terms for SO_4, for example, should be modified to (BH):

$$\left\{ [SO_4]_{L,t} - [SO_4]_{in,t+\Delta t} \right\} \exp \left\{ \frac{-\Delta t}{\tau_t} \right\}$$

4. The values printed at t=0 are really an average over the first time step (BH). I assume this means that the initial condition apply at t=0, but the actual sampling of dose etc. is at t=5 a, 55 a, ...

Return Call from SENES, April 10, 1:46 p.m.

(Bruce Halbert, Mike Murray)

1. Didn't want to "start" calculations with zero concentrations in the lake etc., since the tailings are assumed to have been in place prior to the simulations. For the finite difference equations, a "previous" value is assumed, and sometimes corresponds to the stated initial conditions, but applying at t=- t, and sometimes corresponding to an unstated assumption (see examples below).

   Eq., The initial Ca concentration in lake = background Ca levels at t=-10 a.

2. What about cases where the finite difference equations require values at t=0? For example to calculate RAMIX at t=0, you require CRAAVG at t=0, where CRAAVG does not come from a finite difference equation. Answer: At t=0, a different calculation is done for variables such as:

   - SO_4 from pyrite oxidation (mixing zone) at t=0 is equal to SO_4 concentration from pyrite oxidation at the bottom of the unsaturated zone at t=0. This is equal to SFLUX/NPRCP.

   - Ca concentration calculated using Eq (A.3.9) and the above SO_4 concentration.

   - Assume gypsum is not depleted during this time step. Therefore the mass of reprecipitated gypsum = 0 at time = 0.

   - Assume that concentration of Ra in the solid phase of the upper layer of the unsaturated zone at t=0 does not contain gypsum (for use in the DUST module). This assumption is somewhat inconsistent with the previous assumption, but avoids a large discontinuity in doses from dust.

   - RAMIX = Ra concentration at the bottom of the unsaturated zone at t=0.
   - SO_4, Ca and Ra in the lake. The recursive relationship for Ra in the lake requires previous values for Ra in the lake and lake sediment. A value of 0 at t=- t is used for both. For SO_4 and Ca, previous values are required for SO_4 and Ca in the lake. It is assumed that the values at t=- t are the stated "background" levels.

3. These changes will have a cumulative effect, so that the previous SYVAC3 results will probably be changed somewhat, as well as shifted in time.

4. No problems detected in these finite difference equations. Some problems were encountered in the iterative equations for LAKEPH and FRACRA (?). A modified Newton's method was used.

Call to SENES April 10/86 10:37 a.m.

(Doug Chambers, Bruce Halbert, Mike Murray)

1. Doug suggested it would be desirable to perform a similar contract on the phase III models.

2. We are still having difficulties with the initial conditions for the finite difference equations. At what time are the initial conditions applied, and at what times are the doses calculated? How is the initial time-step handled?

3. Mike Murray will generate some intermediate output, such as concentrations of Ca in the lake as a function of time, and forward them to us. He will also contact us regarding the initial time-step questions.

SENES CONSULTANTS LIMITED

499 McNicoll Ave.
Willowdale, Ontario
Canada M2H 2C9
Tel; (416) 499-5030

30259

14 April 1986

Dr. Bruce Goodwin
Atomic Energy of Canada Limited
Whiteshell Nuclear Research Establishment
Pinawa, Manitoba
ROE 1LO

Re: Questions Pertaining to Results of PHASE II Probabilistic Assessment of Uranium Mill Tailings

Dear Bruce:

Enclosed, for your information, are further detailed results generated using our Phase II tailings model. The results are summarized in Table 1, and a key to the variable names used is presented in Table 2. The results were generated using the nominal input values summarized in Table 1. These are the same input values used to generate the nominal results provided previously. As indicated on Table 2, I have enclosed values for calcium, sulphate and radium concentrations in the unsaturated and mixing zones and the downstream lake.

I hope you find the enclosed information useful. Please call should you have any further questions.

Yours very truly,

SENES CONSULTANTS LIMITED

M. Murray

M.L. Murray, B.Eng.
Environmental Engineer

ls

Encl.

Table 1.
Nominal Value Results
($[Ca^{2+}]$, $[SO_4^{2-}]$, $[Ra^{2+}]$; 500 years)

```
SUMMARY RESULTS

VARIABLE    MEAN      STD DEV    SKEWNESS   KURTOSIS   GEOM MEAN   GEOM SD    MINIMUM    MAXIMUM

U0       .3195E+01  .0000E+00  .0000E+00  .0000E+00  .3195E+01  .1000E+01  .3195E+01  .3195E+01

CU50     .8025E+01  .0000E+00  .0000E+00  .0000E+00  .8025E+01  .1000E+01  .8025E+01  .8025E+01

CU100    .1297E+02  .0000E+00  .0000E+00  .0000E+00  .1297E+02  .1000E+01  .1297E+02  .1297E+02

U150     .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

CU200    .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

U250     .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

CU300    .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

CU350    .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

400      .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

CU450    .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

500      .0000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+01  .1000E+01  .0000E+00  .0000E+00

SU0      .8693E+02  .0000E+00  .0000E+00  .0000E+00  .8693E+02  .1000E+01  .8693E+02  .8693E+02

SU50     .3462E+02  .0000E+00  .0000E+00  .0000E+00  .3462E+02  .1000E+01  .3462E+02  .3462E+02

100      .2142E+02  .0000E+00  .0000E+00  .0000E+00  .2142E+02  .1000E+01  .2142E+02  .2142E+02
```

Table 1. cont'd.

SU150	.2682E+01	.0000E+00	.0000E+00	.0000E+00	.2682E+01	.1000E+01	.2682E+01	.2682E+01
SU200	.8515E+00	.0000E+00	.0000E+00	.0000E+00	.8515E+00	.1000E+01	.8515E+00	.8515E+00
SU250	.2704E+00	.0000E+00	.0000E+00	.0000E+00	.2704E+00	.1000E+01	.2704E+00	.2704E+00
SU300	.8587E-01	.0000E+00	.0000E+00	.0000E+00	.8587E-01	.1000E+01	.8587E-01	.8587E-01
SU350	.2727E-01	.0000E+00	.0000E+00	.0000E+00	.2727E-01	.1000E+01	.2727E-01	.2727E-01
SU400	.8660E-02	.0000E+00	.0000E+00	.0000E+00	.8660E-02	.1000E+01	.8660E-02	.8660E-02
SU450	.2750E-02	.0000E+00	.0000E+00	.0000E+00	.2750E-02	.1000E+01	.2750E-02	.2750E-02
SU500	.8733E-03	.0000E+00	.0000E+00	.0000E+00	.8733E-03	.1000E+01	.8733E-03	.8733E-03
RU0	.2624E+04	.0000E+00	.0000E+00	.0000E+00	.2624E+04	.1000E+01	.2624E+04	.2624E+04
RU50	.6100E+04	.0000E+00	.0000E+00	.0000E+00	.6100E+04	.1000E+01	.6100E+04	.6100E+04
RU100	.2479E+05	.0000E+00	.0000E+00	.0000E+00	.2479E+05	.1000E+01	.2479E+05	.2479E+05
RU150	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
RU200	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
RU250	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
RU300	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
RU350	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
RU400	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
RU450	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
RU500	.5002E+03	.0000E+00	.0000E+00	.0000E+00	.5002E+03	.1000E+01	.5002E+03	.5002E+03
CN0	.3195E+01	.0000E+00	.0000E+00	.0000E+00	.3195E+01	.1000E+01	.3195E+01	.3195E+01
CN50	.8025E+01	.0000E+00	.0000E+00	.0000E+00	.8025E+01	.1000E+01	.8025E+01	.8025E+01
CN100	.1297E+02	.0000E+00	.0000E+00	.0000E+00	.1297E+02	.1000E+01	.1297E+02	.1297E+02
CN150	.1538E+02	.0000E+00	.0000E+00	.0000E+00	.1538E+02	.1000E+01	.1538E+02	.1538E+02
CN200	.1625E+02	.0000E+00	.0000E+00	.0000E+00	.1625E+02	.1000E+01	.1625E+02	.1625E+02
CN250	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.1000E+01	.1000E+01	.0000E+00	.0000E+00
CN300	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.1000E+01	.1000E+01	.0000E+00	.0000E+00
CN350	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.1000E+01	.1000E+01	.0000E+00	.0000E+00
CN400	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.1000E+01	.1000E+01	.0000E+00	.0000E+00
CN450	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.1000E+01	.1000E+01	.0000E+00	.0000E+00
CN500	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.1000E+01	.1000E+01	.0000E+00	.0000E+00
SN0	.8693E+02	.0000E+00	.0000E+00	.0000E+00	.8693E+02	.1000E+01	.8693E+02	.8693E+02
SN50	.3462E+02	.0000E+00	.0000E+00	.0000E+00	.3462E+02	.1000E+01	.3462E+02	.3462E+02

A-4

table 1. cont'd

```
SM150   .1806E+02  .0000E+00  .0000E+00  .0000E+00  .1806E+02  .1000E+01  .1806E+02  .1806E+02
SM2[redacted]    .1710E+02  .0000E+00  .0000E+00  .0000E+00  .1710E+02  .1000E+01  .1710E+02  .1710E+02
SM250   .2704E+00  .0000E+00  .0000E+00  .0000E+00  .2704E+00  .1000E+01  .2704E+00  .2704E+00
SM300   .8587E-01  .0000E+00  .0000E+00  .0000E+00  .8587E-01  .1000E+01  .8587E-01  .8587E-01
SM3[redacted]    .2727E-01  .0000E+00  .0000E+00  .0000E+00  .2727E-01  .1000E+01  .2727E-01  .2727E-01
SM400   .8660E-02  .0000E+00  .0000E+00  .0000E+00  .8660E-02  .1000E+01  .8660E-02  .8660E-02
SM4[redacted]    .2750E-02  .0000E+00  .0000E+00  .0000E+00  .2750E-02  .1000E+01  .2750E-02  .2750E-02
SM500   .8733E-03  .0000E+00  .0000E+00  .0000E+00  .8733E-03  .1000E+01  .8733E-03  .8733E-03
FM0     .2624E+04  .0000E+00  .0000E+00  .0000E+00  .2624E+04  .1000E+01  .2624E+04  .2624E+04
FM5[redacted]    .2851E+04  .0000E+00  .0000E+00  .0000E+00  .2851E+04  .1000E+01  .2851E+04  .2851E+04
FM100   .4510E+04  .0000E+00  .0000E+00  .0000E+00  .4510E+04  .1000E+01  .4510E+04  .4510E+04
FM1[redacted]    .1402E+05  .0000E+00  .0000E+00  .0000E+00  .1402E+05  .1000E+01  .1402E+05  .1402E+05
FM200   .1181E+05  .0000E+00  .0000E+00  .0000E+00  .1181E+05  .1000E+01  .1181E+05  .1181E+05
FM2[redacted]    .9968E+04  .0000E+00  .0000E+00  .0000E+00  .9968E+04  .1000E+01  .9968E+04  .9968E+04
FM[redacted]     .8425E+04  .0000E+00  .0000E+00  .0000E+00  .8425E+04  .1000E+01  .8425E+04  .8425E+04
FM350   .7133E+04  .0000E+00  .0000E+00  .0000E+00  .7133E+04  .1000E+01  .7133E+04  .7133E+04
FM[redacted]     .6052E+04  .0000E+00  .0000E+00  .0000E+00  .6052E+04  .1000E+01  .6052E+04  .6052E+04
FM450   .5147E+04  .0000E+00  .0000E+00  .0000E+00  .5147E+04  .1000E+01  .5147E+04  .5147E+04
FM5[redacted]    .4390E+04  .0000E+00  .0000E+00  .0000E+00  .4390E+04  .1000E+01  .4390E+04  .4390E+04
CL[redacted]     .1793E-03  .0000E+00  .0000E+00  .0000E+00  .1793E-03  .1000E+01  .1793E-03  .1793E-03
CL50    .3004E-03  .0000E+00  .0000E+00  .0000E+00  .3004E-03  .1000E+01  .3004E-03  .3004E-03
CL1[redacted]    .4241E-03  .0000E+00  .0000E+00  .0000E+00  .4241E-03  .1000E+01  .4241E-03  .4241E-03
CL150   .4844E-03  .0000E+00  .0000E+00  .0000E+00  .4844E-03  .1000E+01  .4844E-03  .4844E-03
CL2[redacted]    .5061E-03  .0000E+00  .0000E+00  .0000E+00  .5061E-03  .1000E+01  .5061E-03  .5061E-03
CL[redacted]     .1000E-03  .0000E+00  .0000E+00  .0000E+00  .1000E-03  .1000E+01  .1000E-03  .1000E-03
CL300   .1000E-03  .0000E+00  .0000E+00  .0000E+00  .1000E-03  .1000E+01  .1000E-03  .1000E-03
CL3[redacted]    .1000E-03  .0000E+00  .0000E+00  .0000E+00  .1000E-03  .1000E+01  .1000E-03  .1000E-03
CL400   .1000E-03  .0000E+00  .0000E+00  .0000E+00  .1000E-03  .1000E+01  .1000E-03  .1000E-03
CL4[redacted]    .1000E-03  .0000E+00  .0000E+00  .0000E+00  .1000E-03  .1000E+01  .1000E-03  .1000E-03
CL5[redacted]    .1000E-03  .0000E+00  .0000E+00  .0000E+00  .1000E-03  .1000E+01  .1000E-03  .1000E-03
FL0     .4280E+02  .0000E+00  .0000E+00  .0000E+00  .4280E+02  .1000E+01  .4280E+02  .4280E+02
FL5[redacted]    .5861E+02  .0000E+00  .0000E+00  .0000E+00  .5861E+02  .1000E+01  .5861E+02  .5861E+02
FL100   .9015E+02  .0000E+00  .0000E+00  .0000E+00  .9015E+02  .1000E+01  .9015E+02  .9015E+02
```

table 1. cont'd

RL150	.2810E+03	.0000E+00	.0000E+00	.0000E+00	.2810E+03	.1000E+01	.2810E+03	.2810E+03
RL200	.2516E+03	.0000E+00	.0000E+00	.0000E+00	.2516E+03	.1000E+01	.2516E+03	.2516E+03
RL250	.2130E+03	.0000E+00	.0000E+00	.0000E+00	.2130E+03	.1000E+01	.2130E+03	.2130E+03
RL300	.1800E+03	.0000E+00	.0000E+00	.0000E+00	.1800E+03	.1000E+01	.1800E+03	.1800E+03
RL350	.1524E+03	.0000E+00	.0000E+00	.0000E+00	.1524E+03	.1000E+01	.1524E+03	.1524E+03
RL400	.1293E+03	.0000E+00	.0000E+00	.0000E+00	.1293E+03	.1000E+01	.1293E+03	.1293E+03
RL450	.1099E+03	.0000E+00	.0000E+00	.0000E+00	.1099E+03	.1000E+01	.1099E+03	.1099E+03
RL500	.9370E+02	.0000E+00	.0000E+00	.0000E+00	.9370E+02	.1000E+01	.9370E+02	.9370E+02
SULPRD	.1400E-05	.0000E+00	.0000E+00	.0000E+00	.1400E-05	.1000E+01	.1400E-05	.1400E-05
RHOTAL	.1400E+07	.0000E+00	.0000E+00	.0000E+00	.1400E+07	.1000E+01	.1400E+07	.1400E+07
O2DIFF	.7000E+02	.0000E+00	.0000E+00	.0000E+00	.7000E+02	.1000E+01	.7000E+02	.7000E+02
KPRIME	.5000E-01	.0000E+00	.0000E+00	.0000E+00	.5000E-01	.1000E+01	.5000E-01	.5000E-01
NPRCP	.4000E+00	.0000E+00	.0000E+00	.0000E+00	.4000E+00	.1000E+01	.4000E+00	.4000E+00
FACTCO	.3000E+00	.0000E+00	.0000E+00	.0000E+00	.3000E+00	.1000E+01	.3000E+00	.3000E+00
DISTCO	.5000E-01	.0000E+00	.0000E+00	.0000E+00	.5000E-01	.1000E+01	.5000E-01	.5000E-01
DUNSAT	.1000E+01	.0000E+00	.0000E+00	.0000E+00	.1000E+01	.1000E+01	.1000E+01	.1000E+01
KSUBDR	.8000E-04	.0000E+00	.0000E+00	.0000E+00	.8000E-04	.1000E+01	.8000E-04	.8000E-04
TVOID	.4000E+00	.0000E+00	.0000E+00	.0000E+00	.4000E+00	.1000E+01	.4000E+00	.4000E+00
FRACGP	.5000E-01	.0000E+00	.0000E+00	.0000E+00	.5000E-01	.1000E+01	.5000E-01	.5000E-01
PYRITE	.7300E+03	.0000E+00	.0000E+00	.0000E+00	.7300E+03	.1000E+01	.7300E+03	.7300E+03
UPERTH	.3000E+01	.0000E+00	.0000E+00	.0000E+00	.3000E+01	.1000E+01	.3000E+01	.3000E+01
UO2TON	.1200E+01	.0000E+00	.0000E+00	.0000E+00	.1200E+01	.1000E+01	.1200E+01	.1200E+01
RAINTH	.9000E+00	.0000E+00	.0000E+00	.0000E+00	.9000E+00	.1000E+01	.9000E+00	.9000E+00
THDISS	.2000E-01	.0000E+00	.0000E+00	.0000E+00	.2000E-01	.1000E+01	.2000E-01	.2000E-01
RAINGP	.2000E+00	.0000E+00	.0000E+00	.0000E+00	.2000E+00	.1000E+01	.2000E+00	.2000E+00
CONCRA	.1100E+02	.0000E+00	.0000E+00	.0000E+00	.1100E+02	.1000E+01	.1100E+02	.1100E+02
THFRAC	.3000E+00	.0000E+00	.0000E+00	.0000E+00	.3000E+00	.1000E+01	.3000E+00	.3000E+00
XK1	.4300E+00	.0000E+00	.0000E+00	.0000E+00	.4300E+00	.1000E+01	.4300E+00	.4300E+00
XK3	.2200E-01	.0000E+00	.0000E+00	.0000E+00	.2200E-01	.1000E+01	.2200E-01	.2200E-01
BETA	.1800E+00	.0000E+00	.0000E+00	.0000E+00	.1800E+00	.1000E+01	.1800E+00	.1800E+00
BF	.1600E-04	.0000E+00	.0000E+00	.0000E+00	.1600E-04	.1000E+01	.1600E-04	.1600E-04
MPCH2O	.1500E+02	.0000E+00	.0000E+00	.0000E+00	.1500E+02	.1000E+01	.1500E+02	.1500E+02

table 1. cont'd.

```
FPREC      .1230E+03  .0000E+00  .0000E+00  .0000E+00  .1230E+03  .1000E+01  .1230E+03  .1230E+03
FN[redacted]D!     .3500E+02  .0000E+00  .0000E+00  .0000E+00  .3500E+02  .1000E+01  .3500E+02  .3500E+02
FH[redacted]02     .1000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+00  .1000E+01  .1000E+00  .1000E+00
HSRC       .2000E+01  .0000E+00  .0000E+00  .0000E+00  .2000E+01  .1000E+01  .2000E+01  .2000E+01
UH[redacted]G      .5300E+01  .0000E+00  .0000E+00  .0000E+00  .5300E+01  .1000E+01  .5300E+01  .5300E+01
HDISP      .6000E+02  .0000E+00  .0000E+00  .0000E+00  .6000E+02  .1000E+01  .6000E+02  .6000E+02
DE[redacted]EL     .1400E-01  .0000E+00  .0000E+00  .0000E+00  .1400E-01  .1000E+01  .1400E-01  .1400E-01
FD[redacted]SF     .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00
FDEPSF     .4000E+00  .0000E+00  .0000E+00  .0000E+00  .4000E+00  .1000E+01  .4000E+00  .4000E+00
FD[redacted]SF     .2000E+00  .0000E+00  .0000E+00  .0000E+00  .2000E+00  .1000E+01  .2000E+00  .2000E+00
FDEPSF     .2000E+00  .0000E+00  .0000E+00  .0000E+00  .2000E+00  .1000E+01  .2000E+00  .2000E+00
FD[redacted]ED     .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00
FD[redacted]ED     .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00
FDEPED     .9000E+00  .0000E+00  .0000E+00  .0000E+00  .9000E+00  .1000E+01  .9000E+00  .9000E+00
FD[redacted]ED     .1000E+00  .0000E+00  .0000E+00  .0000E+00  .1000E+00  .1000E+01  .1000E+00  .1000E+00
DHREXP     .1000E+09  .0000E+00  .0000E+00  .0000E+00  .1000E+09  .1000E+01  .1000E+09  .1000E+09
DU[redacted]XP     .2900E+07  .0000E+00  .0000E+00  .0000E+00  .2900E+07  .1000E+01  .2900E+07  .2900E+07
DU[redacted]XP     .5150E+07  .0000E+00  .0000E+00  .0000E+00  .5150E+07  .1000E+01  .5150E+07  .5150E+07
DH[redacted]EXP    .5150E+07  .0000E+00  .0000E+00  .0000E+00  .5150E+07  .1000E+01  .5150E+07  .5150E+07
VE[redacted]AM     .3000E-08  .0000E+00  .0000E+00  .0000E+00  .3000E-08  .1000E+01  .3000E-08  .3000E-08
VEGLAM     .5000E-06  .0000E+00  .0000E+00  .0000E+00  .5000E-06  .1000E+01  .5000E-06  .5000E-06
VE[redacted]AM     .5000E-06  .0000E+00  .0000E+00  .0000E+00  .5000E-06  .1000E+01  .5000E-06  .5000E-06
VE[redacted]AM     .5000E-06  .0000E+00  .0000E+00  .0000E+00  .5000E-06  .1000E+01  .5000E-06  .5000E-06
YDEN1      .5000E+03  .0000E+00  .0000E+00  .0000E+00  .5000E+03  .1000E+01  .5000E+03  .5000E+03
Y[redacted]2       .7500E+03  .0000E+00  .0000E+00  .0000E+00  .7500E+03  .1000E+01  .7500E+03  .7500E+03
YDEN3      .2000E+04  .0000E+00  .0000E+00  .0000E+00  .2000E+04  .1000E+01  .2000E+04  .2000E+04
Y[redacted]4       .2000E+04  .0000E+00  .0000E+00  .0000E+00  .2000E+04  .1000E+01  .2000E+04  .2000E+04
FB[redacted]       .1200E-08  .0000E+00  .0000E+00  .0000E+00  .1200E-08  .1000E+01  .1200E-08  .1200E-08
FACT1      .2300E+00  .0000E+00  .0000E+00  .0000E+00  .2300E+00  .1000E+01  .2300E+00  .2300E+00
FA[redacted]2      .2300E+00  .0000E+00  .0000E+00  .0000E+00  .2300E+00  .1000E+01  .2300E+00  .2300E+00
FACTA      .7500E+00  .0000E+00  .0000E+00  .0000E+00  .7500E+00  .1000E+01  .7500E+00  .7500E+00
FA[redacted]S      .7500E+00  .0000E+00  .0000E+00  .0000E+00  .7500E+00  .1000E+01  .7500E+00  .7500E+00
F[redacted]        .7200E+00  .0000E+00  .0000E+00  .0000E+00  .7200E+00  .1000E+01  .7200E+00  .7200E+00
```

table 1. cont

```
FMEAT   .5000E+00  .0000E+00  .0000E+00  .0000E+00  .5000E+00  .1000E+01  .5000E+00  .5000E+00

FISHIN  .8000E+04  .0000E+00  .0000E+00  .0000E+00  .8000E+04  .1000E+01  .8000E+04  .8000E+04

GF      .6000E+00  .0000E+00  .0000E+00  .0000E+00  .6000E+00  .1000E+01  .6000E+00  .6000E+00

F       .4000E+00  .0000E+00  .0000E+00  .0000E+00  .4000E+00  .1000E+01  .4000E+00  .4000E+00
Stop - Program terminated.

B:\B
```

Table 2

KEY TO RESULTS (TABLE 1)

CU(t)* = $[Ca^{2+}]$ unsaturated zone (mol/m^3)

SU(t) = $[SO_4{}^{2-}]$ unsaturated zone (mol/m^3)

RU(t) = $[Ra^{2+}]$ unsaturated zone (Bq/m^3)

CM(t) = $[Ca^{2+}]$ mixing zone (mol/m^3)

SM(t) = $[SO_4{}^{2-}]$ mixing zone (mol/m^3)

RM(t) = $[Ra^{2+}]$ mixing zone (Bq/m^3)

CL(t) = $[Ca^{2+}]$ lake (mol/L)*

RL(t) = $[Ra^{2+}]$ lake (Bq/m^3)

* all results reported for time (t).
(0, 50, 100....500 years)

Call to SENES May 6/86 9:51 a.m.

(Bruce Halbert)

1. We have localized our remaining difficulties in the UTAP-SYVAC3 comparison to the initial condition for x in Eq. 3.24 of Appendix A. In previous conversations, SENES has stated that x=0 at t=0. We find a better fit if x>0 at t=0.

2. Bruce Halbert noted that the initial condition is probably x=0 at t=-10a. He will check the code once again. If possible, he will run the deterministic version and report back to us the value calculated for x at t=0.

Return call from SENES, May 6/86 10:50 a.m.
(Mike Murray)

1. The initial condition previously stated for X (FRACRA) at t=0 is incorrect. The initial condition is

$$X(0) = X^{o}_{Ra/gyp}$$

where $X^{o}_{Ra/gyp}$ is defined on page A-3 of Appendix A.

APPENDIX B: SPECIFICATIONS FOR THE TAILINGS MODEL

Specifications for the UTAP (Uranium Tailings Assessment Program) model have been developed from the SENES Phase II Report[1] . The specifications consist of three components: data flow diagrams (DFD's), minispecifications (minispecs), and the data dictionary (DD). The following notes describe the format of the DFD's, and minispecs, and are followed by the DFD's and minispecs developed from the uranium tailing model. Notes for the data dictionary are presented in Appendix C.

Data Flow Diagrams (DFD's)

Data flow diagrams have been well described in published references (see for example DeMarco[2] and Gane and Sarson[3] The following definition of a data flow diagram comes from Page-Jones[4] :

> A DFD is a network representation of a system, and shows the active components of the system and the data interfaces between them. It is also known informally as a Bubble Chart, since it consists of little circles resembling bubbles.

The following conventions have been followed in preparing DFD's for these specifications:

(1) Circles (bubbles) represent processes, and arrows represent data flows between processes. Solid arrows represent data flows that are functions of time; dashed arrows represent data flows that are constant values. All bubbles and data flows are labelled. Bubbles are labelled with verb phrases such as "Calculate concentrations from atmospheric pathways", and data flows have short unique names of up to six characters (plus optional subscript) such as CAIR(RA). An asterisk next to the name of a data flow means that the data flow appears more than once in the same DFD. A bubble may have any number of incoming and outgoing data flows. If a bubble has only one outflow, say "pH of Lake", then it is named "Calculate pH of Lake".

---

1)SENES, 1985. Probabilistic model development for the assessment of the long-term effects of uranium mill tailings in Canada - Phase II; Volume 3 - Probabilistic analysis user's manual and documentation for the Uranium Tailings Assessment Program (UTAP); a research report prepared under contract # OSQ84-00207 for The National Uranium Tailings Program, Canada Centre for Mineral and Energy Technology, Energy, Mines and Resources Canada, 555 Booth Street, Ottawa, Ontario, K1A 0G1

2)DeMarco, Tom, 1979. Structured analysis and system specification; YOURDON Inc., New York

3)Gane, Chris and Sarson, Trish, 1979. Structured systems analysis: Tools and techniques; Prentice-Hall Inc., Englewood Cliffs, New Jersey.

4)Page-Jones, M., 1980. The practical guide to structured systems design; YOURDON Inc., New York, p.59.

Otherwise, a more general name is used. Bubbles are also labelled with unique numeric identifiers, like "1.3.2" (the meaning of the sequence of numbers is explained below).

(2) The complete UTAP model is represented by a set of DFD's. The set contains one DFD called Figure 0 to represent the entire system at a general level. Each bubble in Figure 0 may be expanded into a full DFD representing more detailed data flows and processes. Bubbles appearing on the detailed DFD's may themselves be expanded into more detailed DFD's, recursively, until the processes are simple enough to be described by a minispec (see the next section). The bubbles are numbered to show the successive expansions. Bubbles shown in Figure 0 are numbered 1, 2, 3, etc. Bubbles shown in Figure 2, which is an expansion of bubble 2, are numbered 2.1, 2.2, 2.3, etc. Those in Figure 2.3, an expansion of bubble 2.3, are numbered 2.3.1, 2.3.2, 2.3.3, etc. For example, the bubble 2.2 is called "Calculate Rn gas emanation from the tailings zone", which is one sub-process in Figure 2. Figure 2 is an expansion of bubble 2 in Figure 0, and both bubble 2 and Figure 2 are labelled "Calculate concentrations from atmospheric pathways".

(3) All variables needed in a process are shown as data flows into the appropriate bubble. All variables that are explicitly calculated must have a source bubble from which they are shown emerging. In contrast, sampled parameters are never shown emerging from a bubble; they are implicitly assumed to be calculated by the SYVAC or UTAP sampling routines. Therefore, data flows shown coming from the edge of a DFD are either sampled parameters, or calculated variables that were generated on a higher level diagram. The top DFD, Figure 0, shows all the sampled parameters that are used in the model. To prevent clutter of Figure 0, these parameters are grouped together in lists. For example, bubble 2 (Calculate concentrations from atmospheric pathways) requires the list of parameters called PATMOS. Where such a list is used, a note is placed at the bottom of the figure expanding the list (key words like PATMOS are also defined in the data dictionary).

Minispecifications

According to DeMarco[5] a minispecification (minispec) is "a concise statement of user policy rules governing a given elemental function". Minispecs detail the workings of each process bubble that is not expanded into a DFD. Due to the detailed mathematical nature of the UTAP tailings model, two standard formats for minispecs have been chosen: one that lends itself to specification of the technical functions which are part of the uranium tailings model, and one that lends itself to specification of a more general nature.

---

5) DeMarco, Tom, 1982. Controlling software projects; YOURDON Inc., New York.

The type of minispec used for most of the specifications of the uranium tailings model is known as a data dependency diagram, and it contains the following components:

- A diagram showing data dependencies: Data dependency diagram is defined by Smith[6] to be "an unambiguous depiction of dependencies between data fields". As modified for this application, a DDD comprises a labelled ellipse for every calculated variable and a labelled rectangle for every sampled parameter, joined by arrows showing which variables depend on which. The labels are the long and short names of the appropriate variables. Ellipses also include the number of any DFD bubble used to calculate the variable. Since the DDD is a prominent part of the minispec, the entire minispec is often called a DDD.

- Definitions of the outgoing data flows: All data flows that are calculated in a process and used elsewhere are defined at the top of the minispec. The definition is the same as that used in the data dictionary (see Appendix C), except that some background information is suppressed. The definition comprises a short name, a long name, SI units, and a text definition. Whenever possible, the short name corresponds to the variable name used in UTAP and the variable name used in SYVAC3. The short name also appears as an entry in the data dictionary.

- Equations: Equations are given for any calculations performed in the minispec. These equations are always referenced back to the SENES documentation, or to more recent corrections. Where SENES documentation uses symbols to refer to variables, a table of equivalences is given mapping symbols to variable names.

- Units check: Wherever possible, the units of equations are checked to ensure consistency. Several errors in the SENES documentation were found by the units check. The units are shown in the minispec to show that the check was performed.

- Notes: Any additional explanation of the minispec is given at the bottom in the form of numbered notes.

The second type of minispec used here is more suited for the specification of general-purpose processes or algorithms, and is known as a module interface specification. It has the following components:

- Interface specification: The interface specification as described by Page-Jones[7] "allow you to define the function of a module without delving into excessive detail. You tell the programmer of the module

---

6) Smith, Henry C., 1985-August. Database design: Composing fully normalized tables from a rigorous dependency diagram; Communications of the ACM, 28(8), pp. 826-838.

7) Page-Jones, M., 1980. The practical guide to structured systems design; YOURDON Inc., New York, pp 91-92.

what input you promise to provide when you call his module, and what output you expect when his module returns, and the functions you expect his module to carry out. The functions should be stated in a simple sentence that establishs relationships between the modules's inputs and outputs." As used here, the module interface specifications include written descriptions of the input and output, and a written description of the method to be employed.

- Sample tests: To augment the above information, sample tests that must be performed are provided. The sample tests include typical input and the expected output.

LIST OF FIGURES OF THE DATA FLOW DIAGRAMS AND DATA DEPENDENCY DIAGRAMS FOR THE TAILINGS MODEL

Figure No.	Title	Page
0	Overview Data Flow Diagram (DFD) - NUTP Tailings Model	B-7
1.0	Calculate Tailings Source Terms	B-8
1.1	Calculate Initial Gypsum Content in Tailings	B-9
1.2	Model Chemistry of Ca and SO4	B-10
1.2.1	Calculate Chemical Constants	B-11
1.2.1.1	Calculate Sulphate Flux from Oxd'n of Pyrite	B-12
1.2.1.2	Calculate Init. Conc. of Sulphate at Bottom of Unsat. Zone	B-13
1.2.1.3	Calculate Rate of Oxd'n of Pyrite	B-14
1.2.1.4	Calculate Sq. Root of Eqbm. Const.	B-15
1.2.2	Calculate Time to Deplete Gypsum in Unsat. Zone	B-16
1.2.3	Calculate Mixing Zone Concentrations	B-17
1.2.3.1	Calculate Conc. of S04 Aq. in the Unsat. Zone	B-18
1.2.3.2	Calculate Conc. of Ca Aq. in the Unsat. Zone	B-19
1.2.3.3	Calculate Conc. of Ions from Gyp. Diss'n in Mixing Zone	B-20
1.2.3.4	Calculate Mass of Gypsum Left in the Mixing Zone	B-21
1.2.3.5	Calculate Conc. of S04 Aq. in the Mixing Zone	B-22
1.2.3.6	Calculate Conc. of Ca Aq. in the Mixing Zone	B-23
1.2.4	Calculate Position of Gypsum Depletion Front	B-24
1.2.4.1	Calculate Depth of Upper Unsat. Zone	B-25
1.2.4.2	Calculate Incremental Change in DEPTH1 from TIME to TIME+DELT	B-26
1.2.4.3	Calculate Depth of Lower Unsat. Zone	B-27
1.2.5	Calculate Reprec. Gypsum Mass in Lower Unsat. Zone	B-28
1.3	Calculate Ra Conc. in Upper Unsat. Zone	B-29
1.3.1	Calculate Total Init. Conc. of Ra in Solids	B-30
1.3.2	Calculate Init. Conc. of Ra in Gypsum	B-31
1.3.3	Calculate Init. Solid Phase Conc. of Th-230	B-32
1.3.4	Calculate Init. Conc. of Immobile Ra in Tailings	B-33
1.3.5	Calculate Conc. of Ra-226 Due to Ingrowth from Th-230	B-34
1.3.6	Calculate Ra Conc. in Upper Unsat. Zone	B-35
1.3.7	Calculate Conc. of Ra Aq. from Th Decay in Unsat. Zone	B-36
1.3.7.1	Calculate Init. Solid Phase Conc. of Th-232	B-37
1.3.7.2	Calculate Init. Total Th in Solid Phase	B-38
1.3.7.3	Calculate MF of Th-230 to Th-232	B-39
1.3.7.4	Calculate Total Th in Solid Phase	B-40
1.3.7.5	Calculate Conc. of Ra Aq. from Th Decay in Unsat. Zone	B-41
1.4	Calculate Radium Reprecipitation	B-42
1.5	Calculate Ra Conc. in Mixing Zone	B-44
1.5.1	Calculate Avg. Conc. of Ra to Mixing Zone	B-45
1.5.2	Calculate Conc. of Ra Aq. in Mixing Zone	B-47
1.6	Calculate Ra Conc. in Lower Unsat. Zone	B-48
2.0	Calculate Concentrations from Atmospheric Pathways	B-49
2.1	Calculate Emission Rate of Ra in Dust from Tailings Surface	B-50

2.2 Calculate Rn Gas Emanation from the Tailings Source B-51
2.2.1 Calculate Rn Emission Rate B-52
2.2.2 Calculate Emission of Rn from the Two Unsat. Layers B-53
2.2.3 Calculate the Reciprocal of Rn Diffusion Path Length B-54
2.3 Calculate Concentrations in Air at Receptor B-55
2.4 Calculate Concentrations in Vegetables B-56
2.5 Calculate Concentrations in Caribou B-57

3.0 Calculate Dose B-58
3.1 Calculate Total Dose B-59
3.2 Calculate Dose from Fish B-60
3.3 Calculate Dose from Water B-61
3.4 Calculate Dose from Vegetation B-62
3.5 Calculate Dose from Caribou B-63
3.6 Calculate Dose from Rn Gas B-64
3.7 Calculate Dose from Ra in Dust B-65

4.0 Calculate Concentrations from Lake Pathways B-66
4.1 Calculate Lake Concentration and Retention Time B-67
4.1.1 Calculate Ra Concentration in Lake B-68
4.1.2 Calculate Mass of Ra in Lake Water B-69
4.1.3 Calculate Retention Time in Lake B-71
4.2 Calculate Ra Concentration in Fish B-72
4.3 Calculate Lake pH B-73
4.3.1 Calculate pH of Lake B-74
4.3.2 Calculate Conc. of Ca in Lake B-75
4.3.3 Calculate Conc. of S04 in Lake B-76
4.3.4 Calculate Inflow of Ca (and S04) to Lake B-77

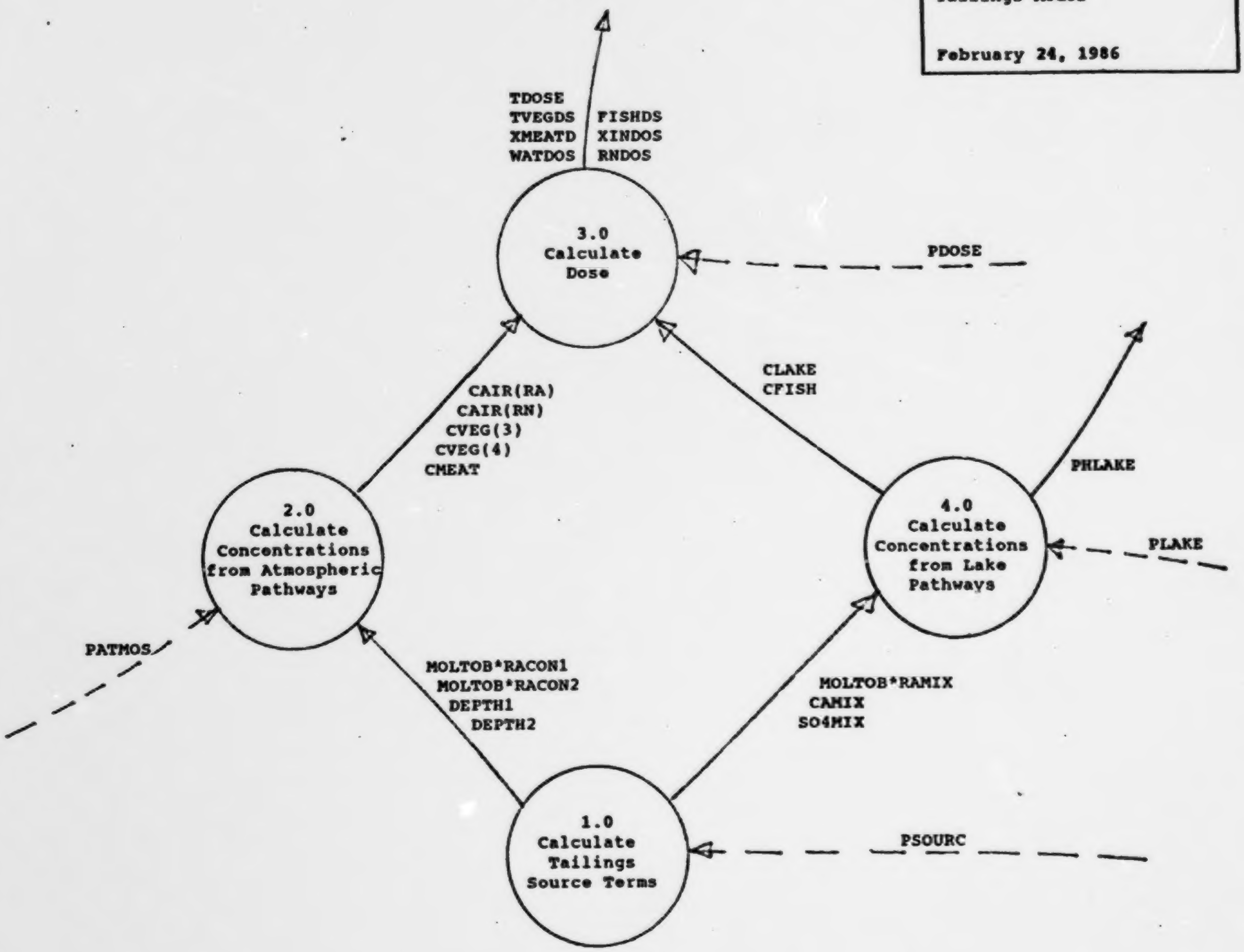
LIST OF FIGURES FOR THE MODULE INTERFACE SPECIFICATIONS

Figure No. **Title** **Page**

5.0 Module Interface Specification for BISECT B-78

Figure 0
Overview of NUTP
Tailings Model

February 24, 1986


TDOSE
TVEGDS
XMEATD
WATDOS
FISHDS
XINDOS
RNDOS
3.0 Calculate Dose
PDOSE
CAIR(RA)
CAIR(RN)
CVEG(3)
CVEG(4)
CMEAT
CLAKE
CFISH
PHLAKE
2.0 Calculate Concentrations from Atmospheric Pathways
4.0 Calculate Concentrations from Lake Pathways
PLAKE
PATMOS
MOLTOB*RACON1
MOLTOB*RACON2
DEPTH1
DEPTH2
MOLTOB*RAMIX
CAMIX
SO4MIX
1.0 Calculate Tailings Source Terms
PSOURC

Note: PATMOS, PDOSE, PLAKE and PSOURC are lists of input (sampled) parameters required by the bubbles they point to. They contain the following parameters:

PATMOS = Contents of PAIRDS + PDUST + PRADON + PVGDEP + PVGEAT (these lists are defined in Figure 2.0)

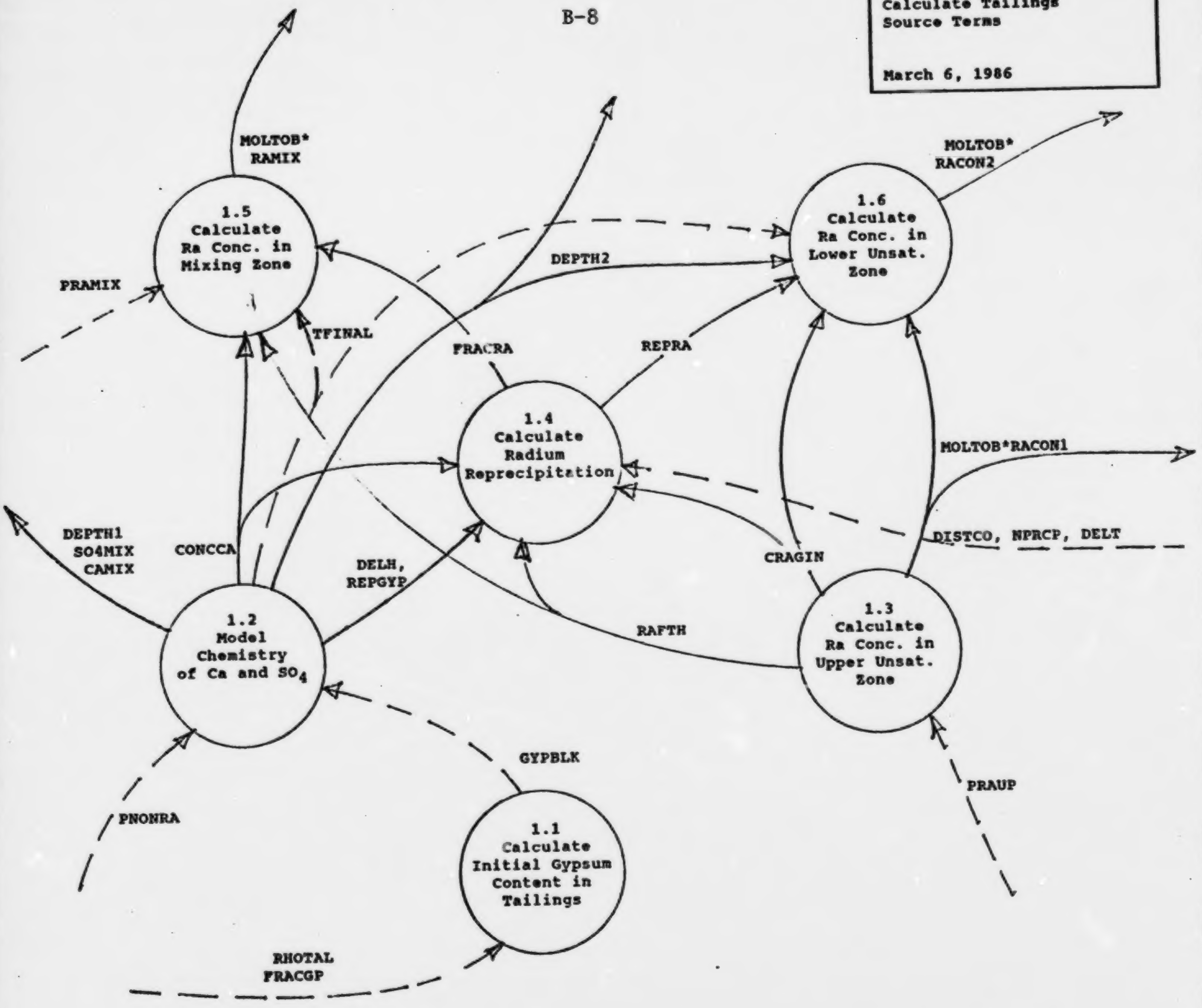
PDOSE = DCFING, DCFINH, FFISH, FISHIN, WATIN, FWAT, VEGIN(v), FLOC (v), FRACV(v) (v=3,4), XMEATI, FMEAT, F, OF, XINHIN.

PLAKE = ATAIL, XK1, XK3, BETA, NPRCP, VLAKE, AWAT, ACTCO, BF. The variable DELT is also included, although it is not a sampled parameter.

PSOURC = RAINGP, RAINTH, THFRAC, THLAM, RALAM, RNLAM, RHOTAL, CONCRA, DUNSAT, NPRCP, FRACGP, KSUBSO, FACTCO, PYRITE, O2CONC, O2DIFF, KPRIME, SULPRD, DISTCO, UO2TON, UPERTH, THDISS, DMIX, TVOID, KSUBDR. The variable DELT is also included, although it is not a sampled parameter.

Figure 1.0
Calculate Tailings
Source Terms

March 6, 1986


MOLTOB*
RAMIX
1.5 Calculate Ra Conc. in Mixing Zone
PRAMIX
TFINAL
DEPTH2
FRACRA
REPRA
MOLTOB*
RACON2
1.6 Calculate Ra Conc. in Lower Unsat. Zone
1.4 Calculate Radium Reprecipitation
MOLTOB*RACON1
DEPTH1
SO4MIX
CAMIX
CONCCA
DELH,
REPGYP
CRAGIN
DISTCO, NPRCP, DELT
1.2 Model Chemistry of Ca and SO$_4$
RAFTH
1.3 Calculate Ra Conc. in Upper Unsat. Zone
GYPBLK
PRAUP
PNONRA
1.1 Calculate Initial Gypsum Content in Tailings
RHOTAL
FRACGP

Note: 1. PRAMIX, PRAUP and PNONRA are clusters of input (sampled) parameters that are used by the bubbles they point to. Their contents are:

PRAMIX = DISTCO + DMIX + RHOTAL + DELT + KSUBDR + NPRCP + TVOID

PRAUP = CONCRA + RAINGP + RALAM + RHOTAL + PRAFTH

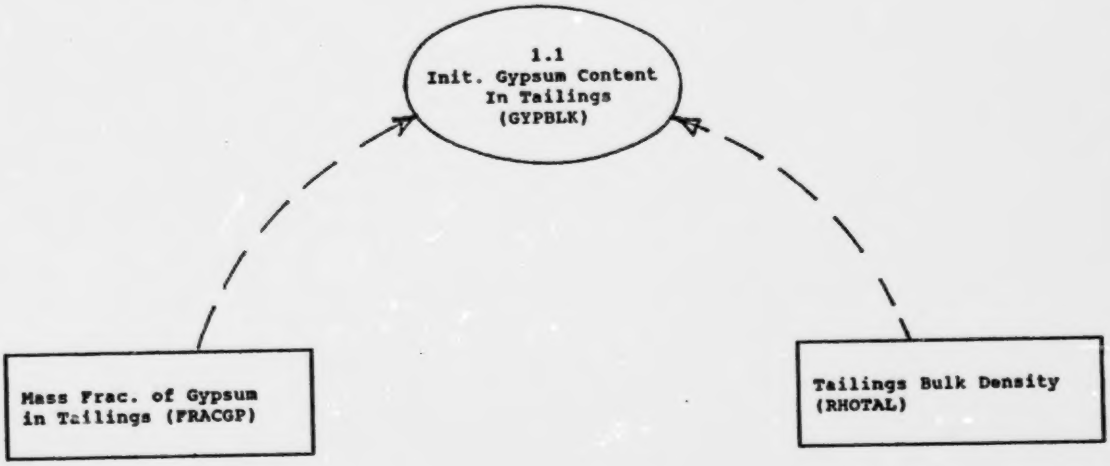
PNONRA = DUNSAT + NPRCP + DELT + PCHEM + PMIXZ

2. PCHEM, PMIXZ and PRAFTH are themselves clusters of parameters defined in the appropriate figures.

Figure 1.1
Calculate Initial
Gypsum Content in
Tailings (GYPBLK)

March 7, 1986

GYPBLK INIT. GYPSUM CONTENT IN TAILINGS [mol m^{-3}]. Initial solid phase gypsum content of tailings.


1.1
Init. Gypsum Content
In Tailings
(GYPBLK)
Mass Frac. of Gypsum
in Tailings (FRACGP)
Tailings Bulk Density
(RHOTAL)

$$\text{GYPBLK} = \frac{\text{RHOTAL} * \text{FRACGP}}{136} \qquad [\text{mol m}^{-3}] \qquad (\text{Eq. A.2.1})$$

Unit check: $\text{g m}^{-3} * (-) / [\text{g mol}^{-1}] = [\text{mol m}^{-3}]$

Note: 1. The constant (136) has units of [g mol^{-1}] and corresponds to the molecular weight of $CaSO_4$. This may be in error, since gypsum is $CaSO_4 \cdot 2H_2O$.

2. Symbols used in Appendix A.

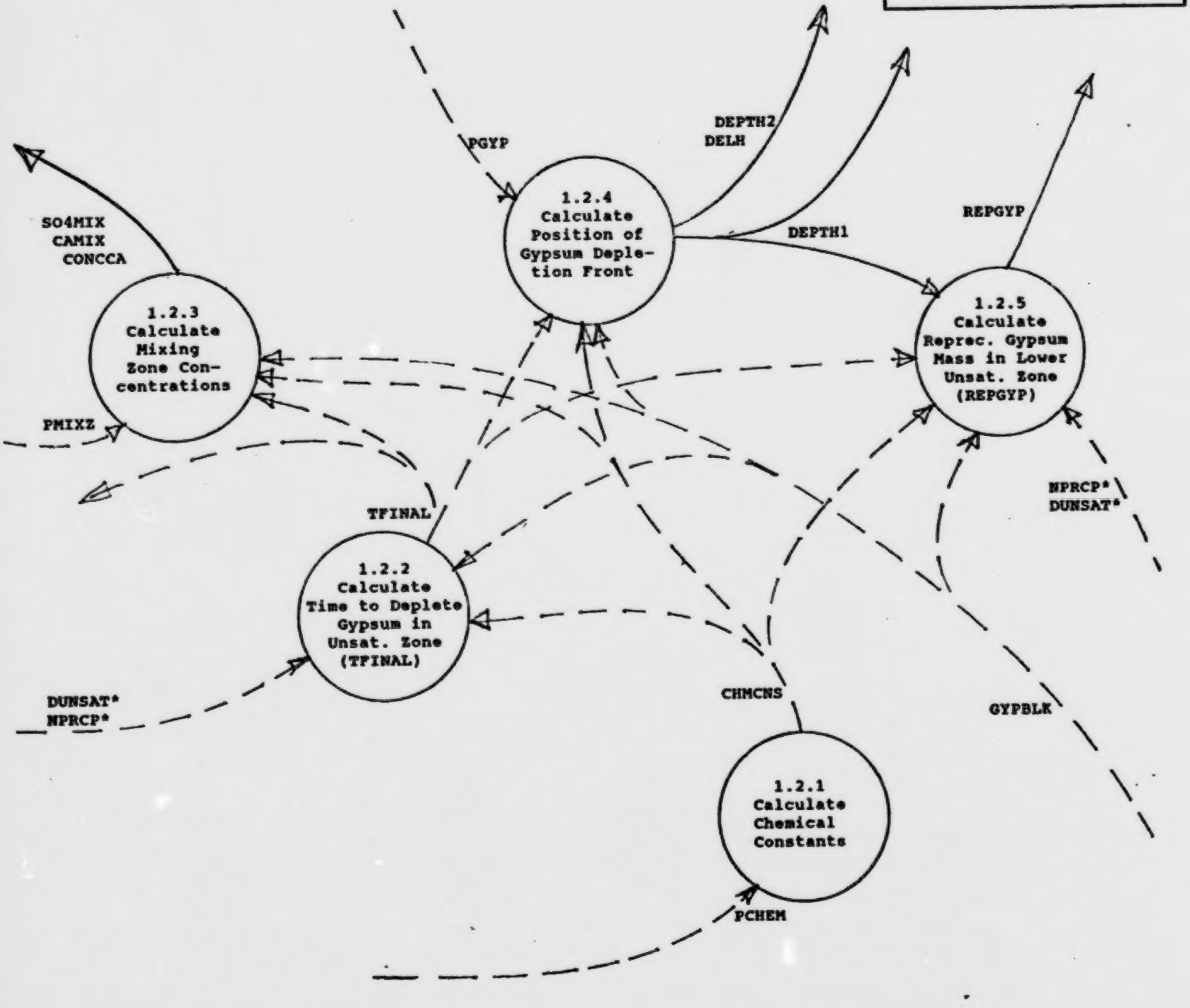
GYPBLK = g_o

FRACGP = x_{gyp}

RHOTAL = ρ_T

Figure 1.2
Model Chemistry of
Ca and SO_4

March 6, 1986


PGYP
DEPTH2
DELH
1.2.4
Calculate
Position of
Gypsum Deple-
tion Front
DEPTH1
REPGYP
SO4MIX
CAMIX
CONCCA
1.2.3
Calculate
Mixing
Zone Con-
centrations
1.2.5
Calculate
Reprec. Gypsum
Mass in Lower
Unsat. Zone
(REPGYP)
PMIXZ
NPRCP*
DUNSAT*
TFINAL
1.2.2
Calculate
Time to Deplete
Gypsum in
Unsat. Zone
(TFINAL)
DUNSAT*
NPRCP*
CHMCNS
GYPBLK
1.2.1
Calculate
Chemical
Constants
PCHEM

Note: 1. PMIXZ, PGYP and PCHEM are clusters of input (sampled) parameters, consisting of:
PCHEM = DUNSAT + FACTCO + KPRIME + KSUBSO + O2CONC + O2DIFF + NPRCP + PYRITE + RHOTAL + SULPRD
PMIXZ = DELT + DMIX + DUNSAT + FACTCO + KSUBSO + NPRCP
PGYP = NPRCP + DUNSAT + DELT

2. CHMCNS is a cluster of calculated variables, consisting of:
CHMCNS = AEQBM + KRATE + CSINIT

Figure 1.2.1
Calculate
Chemical
Constants

March 6, 1986

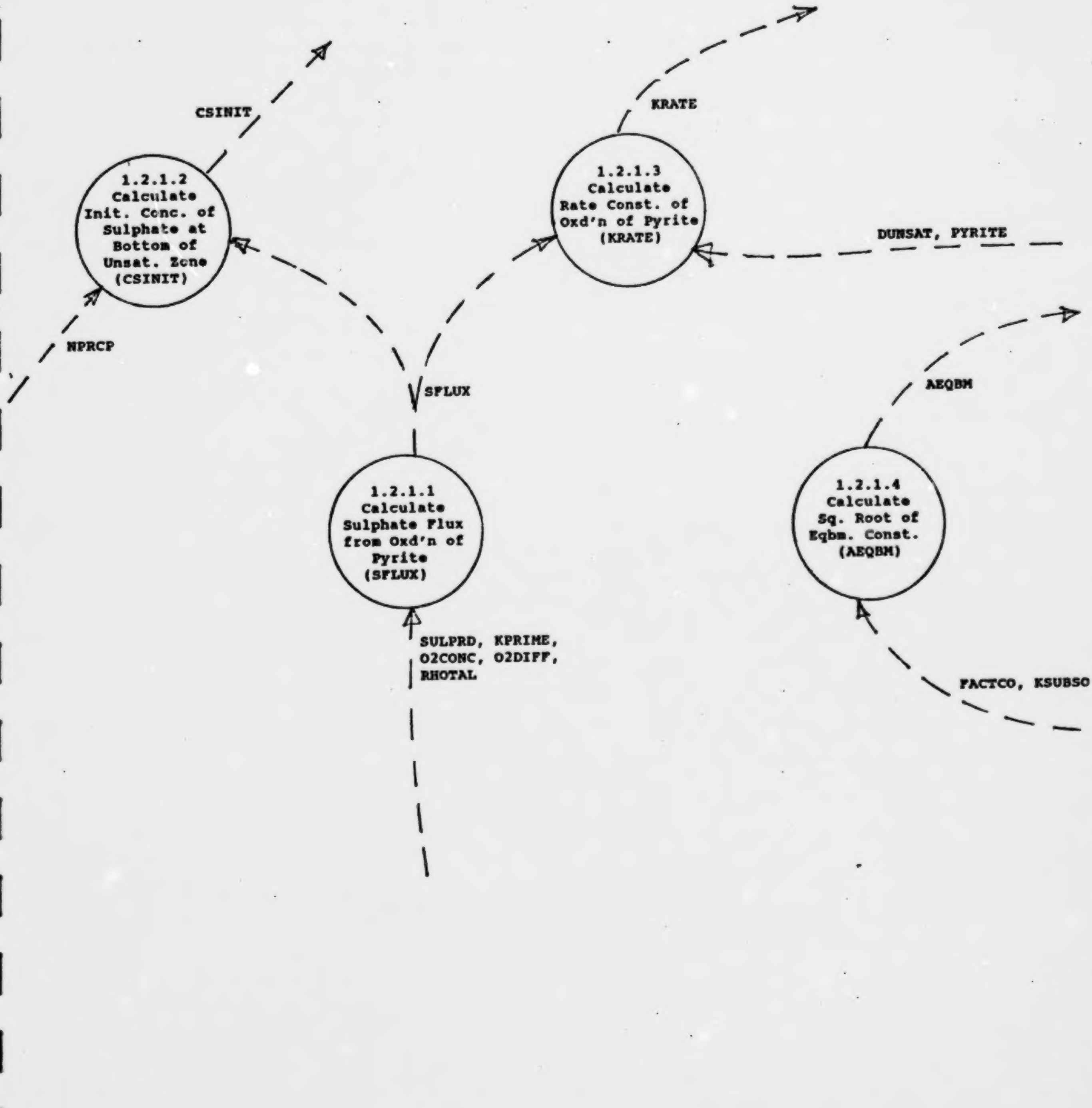
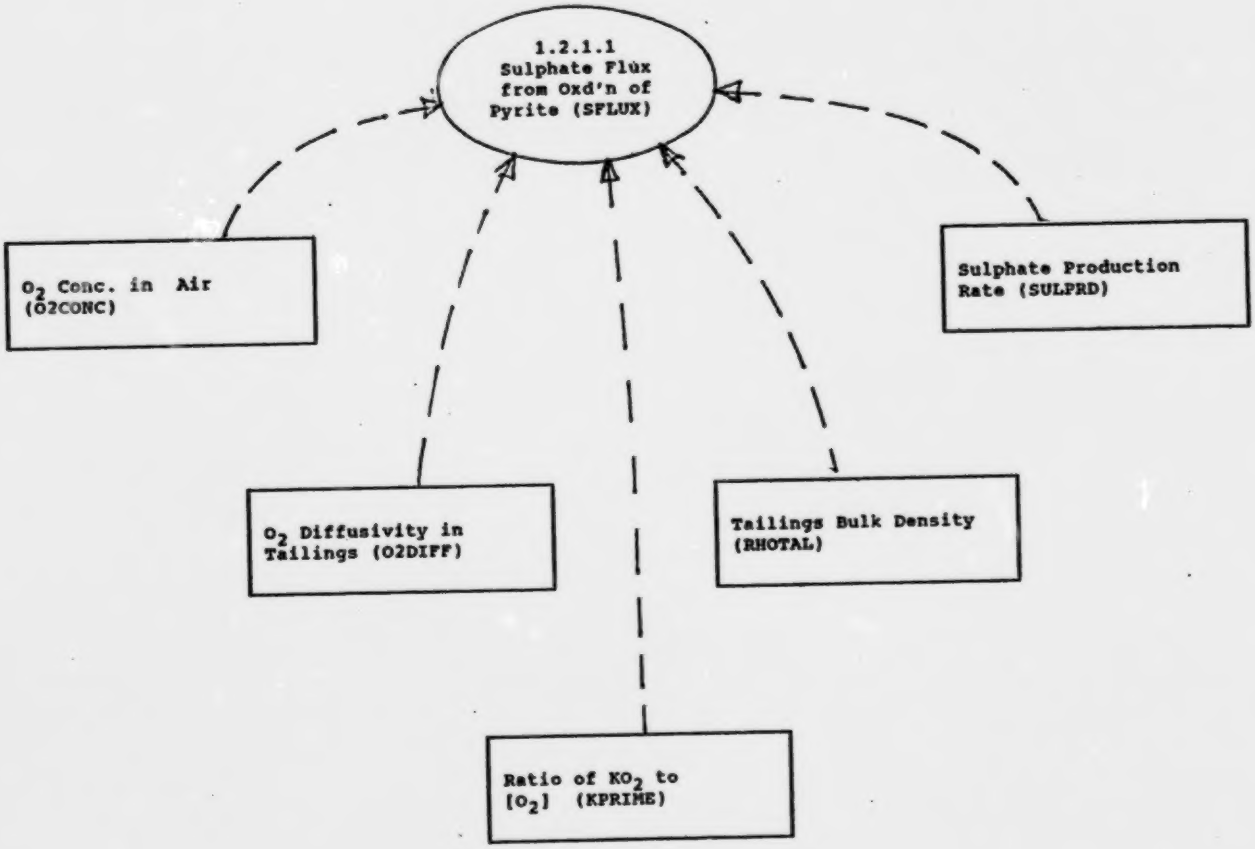

CSINIT
KRATE
1.2.1.2
Calculate
Init. Conc. of
Sulphate at
Bottom of
Unsat. Zone
(CSINIT)
1.2.1.3
Calculate
Rate Const. of
Oxd'n of Pyrite
(KRATE)
DUNSAT, PYRITE
NPRCP
AEQBM
SFLUX
1.2.1.1
Calculate
Sulphate Flux
from Oxd'n of
Pyrite
(SFLUX)
1.2.1.4
Calculate
Sq. Root of
Eqbm. Const.
(AEQBM)
SULPRD, KPRIME,
O2CONC, O2DIFF,
RHOTAL
FACTCO, KSUBSO

Figure 1.2.1.1
Calculate Sulphate Flux from Oxd'n of Pyrite (SFLUX)

March 7, 1986

SFLUX SULPHATE FLUX FROM OXD'N OF PYRITE [mol m^{-2} a^{-1}]. Net sulphate flux out of the bottom of the unsaturated zone (and into the mixing zone) due to the oxidation of pyrite and the downward movement of water.


1.2.1.1 Sulphate Flux from Oxd'n of Pyrite (SFLUX)
O_2 Conc. in Air (O2CONC)
Sulphate Production Rate (SULPRD)
O_2 Diffusivity in Tailings (O2DIFF)
Tailings Bulk Density (RHOTAL)
Ratio of KO_2 to $[O_2]$ (KPRIME)

$$\text{SFLUX} = \left\{ \frac{8}{7.5} * \text{SULPRD} * \text{RHOTAL} * \text{O2CONC} * \text{O2DIFF} * \left[1-K' * \ln\left(\frac{1+K'}{K'}\right)\right]\right\}^{\frac{1}{2}} \quad [\text{mol. } m^{-2}\ a^{-1}] \quad (\text{Eq. A.3.12})$$

Unit check: $[\text{mol } g^{-1}\ a^{-1} * g\ m^{-3} * \text{mol } m^{-3} * m^{2}\ a^{-1} * (-)]^{\frac{1}{2}} = [\text{mol } m^{-2}\ a^{-1}]$

Note: Symbols used in Appendix A

KPRIME = K'

O2CONC = $[O_2]$

O2DIFF = D_{O2}

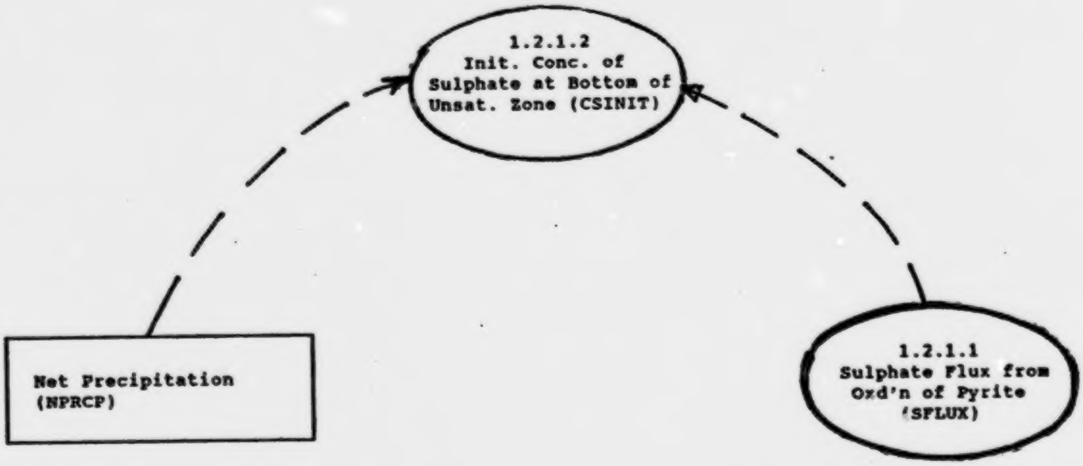
SFLUX = J_s

SULPRD = $\bar{R}_s$

RHOTAL = ρ_T

Figure 1.2.1.2
Calculate Init. Conc.
of Sulphate at Bottom
of Unsat. Zone
(CSINIT)
April 11, 1986

CSINIT INIT. CONC. OF SULPHATE AT BOTTOM OF UNSAT. ZONE [mol m^{-3}]. Initial concentration of sulphate ions in the aqueous phase at the bottom of the unsaturated zone (ie. at the depth DUNSAT).


1.2.1.2
Init. Conc. of
Sulphate at Bottom of
Unsat. Zone (CSINIT)
Net Precipitation
(NPRCP)
1.2.1.1
Sulphate Flux from
Oxd'n of Pyrite
(SFLUX)

$$\text{CSINIT} = \frac{\text{SFLUX}}{\text{NPRCP}} \qquad [\text{mol. } m^{-3}] \qquad (\text{P. A-13})$$

Unit check: mol m^{-2} a^{-1} / [m a^{-1}] = [mol m^{-3}]

Notes: 1. Symbols used in Appendix A:

CSINIT = C_s^o
NPRCP = P
SFLUX = J_s

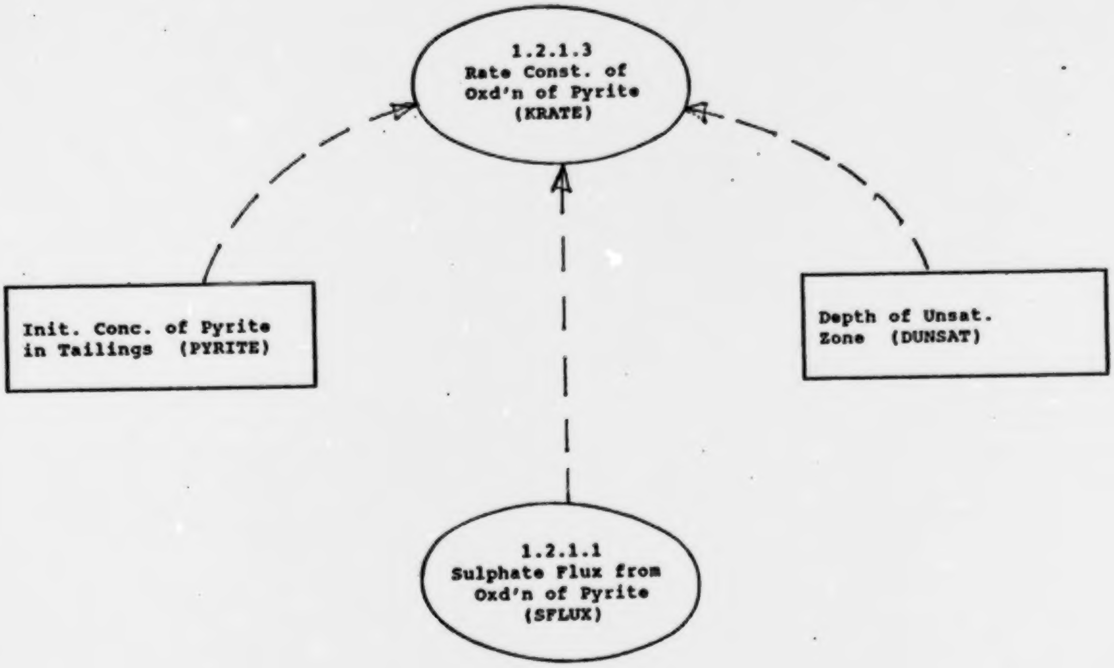
2. It is assumed that this equation applies at t = 0 as noted in a conversation with SENES on 10/4/86.

3. Note that CSINIT only accounts for SO_4 from the oxidation of pyrite.

Figure 1.2.1.3
Calculate Rate Const.
of Oxd'n of Pyrite
(KRATE)

March 6, 1986

KRATE RATE CONST. OF OXD'N OF PYRITE $[a^{-1}]$. Rate constant for oxidation of pyrite.


1.2.1.3
Rate Const. of
Oxd'n of Pyrite
(KRATE)
Init. Conc. of Pyrite
in Tailings (PYRITE)
Depth of Unsat.
Zone (DUNSAT)
1.2.1.1
Sulphate Flux from
Oxd'n of Pyrite
(SFLUX)

$$\text{KRATE} = \frac{\text{SFLUX}}{2 * \text{DUNSAT} * \text{PYRITE}} \qquad [a^{-1}] \qquad \text{(Eq. A.3.13)}$$

Unit check: $\text{mol m}^{-2} \text{ a}^{-1} / [\text{m} * \text{mol m}^{-3}] = [a^{-1}]$

Note: Symbols used in Appendix A:

KRATE = k

DUNSAT = λ_w

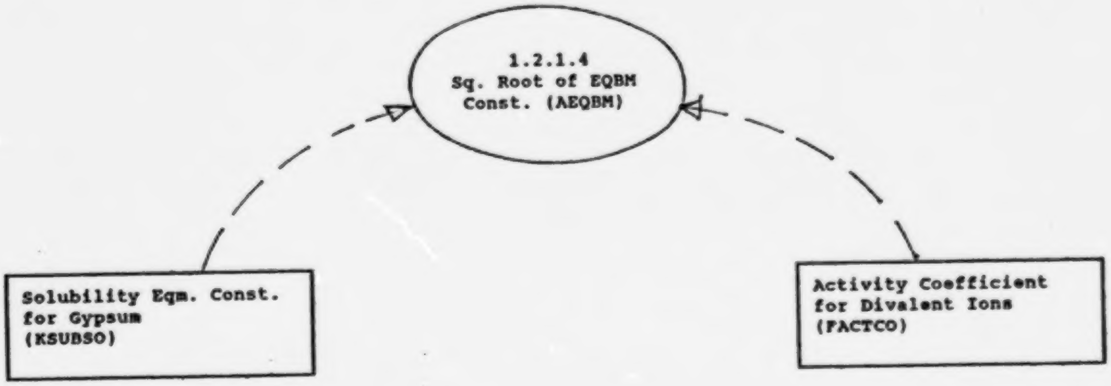
SFLUX = J_s

PYRITE = $[FeS_2]^o$

Figure 1.2.1.4
Calculate Sq. Root of
Eqbm. Const.
(AEQBM)

March 6, 1986

AEQBM SQ. ROOT OF EQBM CONST. [mol m^{-3}]. AEQBM is an intermediate parameter, equal to the square root of the equilibrium constant for the dissolution solubility of gypsum, divided by the activity coefficient.


1.2.1.4
Sq. Root of EQBM
Const. (AEQBM)
Solubility Eqm. Const.
for Gypsum
(KSUBSO)
Activity Coefficient
for Divalent Ions
(FACTCO)

$$AEQBM = \frac{KSUBSO^{1/2}}{FACTCO} \qquad [mol.\ m^{-3}] \qquad (P.\ A\text{-}16)$$

Unit check: $(mol^2\ m^{-6})^{1/2}\ /\ (-) = [mol\ m^{-3}]$

Note: Symbols used in Appendix A:

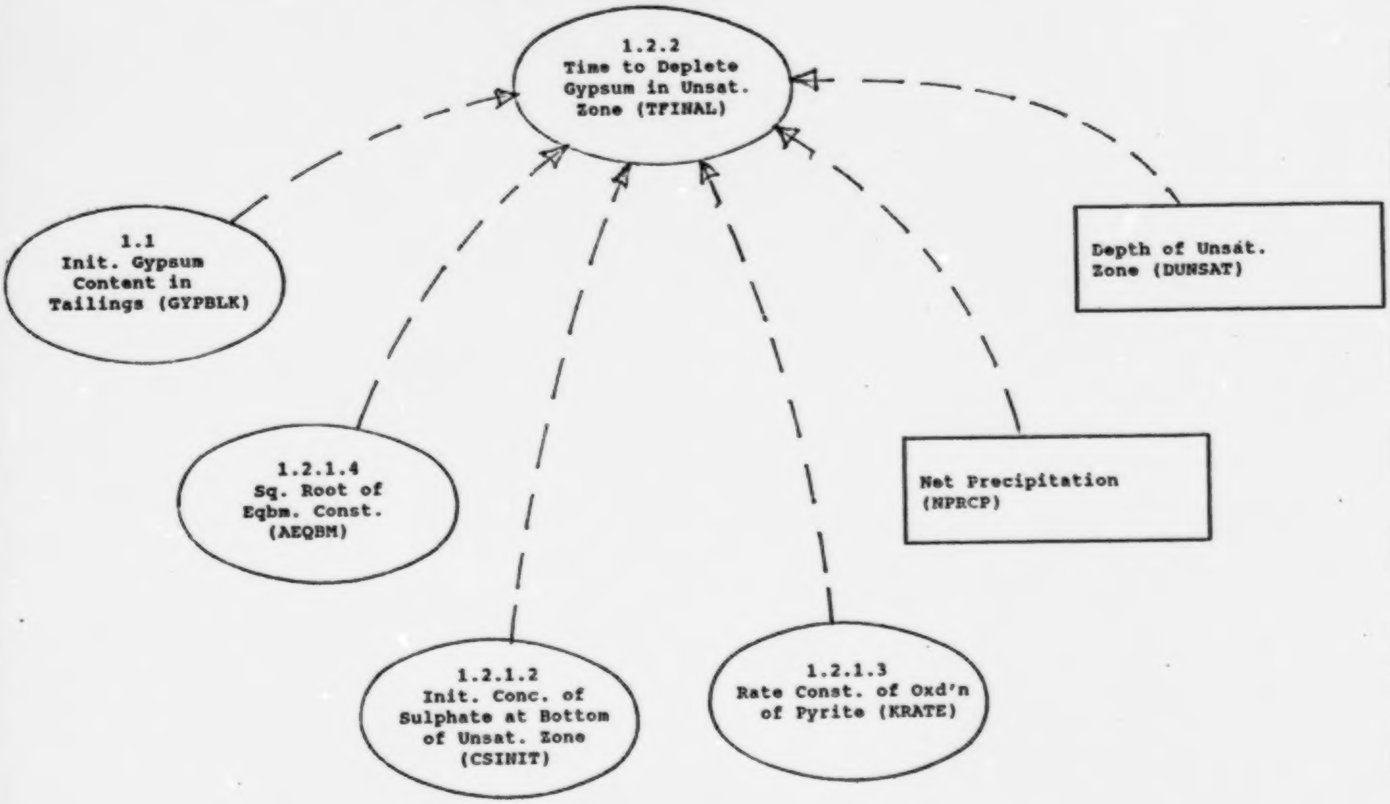
AEQBM = a

KSUBSO = $K^{o}_{s,CaSO4}$

FACTCO = f

Figure 1.2.2
Calculate Time to
Deplete Gypsum in
Unsat. Zone
(TFINAL)
March 7, 1986

TFINAL TIME TO DEPLETE GYPSUM IN UNSAT. ZONE [a]. Time at which the entire unsaturated zone is depleted of gypsum (and when DEPTH1 = DUNSAT).


1.2.2
Time to Deplete
Gypsum in Unsat.
Zone (TFINAL)
1.1
Init. Gypsum
Content in
Tailings (GYPBLK)
Depth of Unsat.
Zone (DUNSAT)
1.2.1.4
Sq. Root of
Eqbm. Const.
(AEQBM)
Net Precipitation
(NPRCP)
1.2.1.2
Init. Conc. of
Sulphate at Bottom
of Unsat. Zone
(CSINIT)
1.2.1.3
Rate Const. of Oxd'n
of Pyrite (KRATE)

Solve the following equation for t_f (TFINAL) iteratively:

$$g_o \lambda_w = Pat_f + \frac{P}{k}\left\{[a^2 + R^2]^{1/2} - [a^2 + R^2 * \mathrm{EXP}(-2kt_f)]^{1/2}\right\} - \frac{PR}{k}[1-\mathrm{EXP}(-kt_f)] - \frac{Pa}{k}\ln\left\{\frac{a + [a^2 + R^2]^{1/2}}{a + [a^2 + R^2 * \mathrm{EXP}(-2kt_f)]^{1/2}}\right\}$$

(Eq. A.3.21)

Initial guess: $t_f = \dfrac{g_o \lambda_w}{Pa}$ (B. Goodwin)

Unit check: $\mathrm{mol\ m^{-3} * m = m\ a^{-1} * mol\ m^{-3} * a} + \dfrac{\mathrm{m\ a^{-1}}}{\mathrm{a^{-1}}}\left\{[\mathrm{mol^2 m^{-6}}]^{1/2} - [\mathrm{mol^2\ m^{-6}}]^{1/2}\right\}$

$- \dfrac{\mathrm{m\ a^{-1}}}{\mathrm{a^{-1}}} * \mathrm{mol\ m^{-3}} * [-] - \dfrac{\mathrm{m\ a^{-1}}}{\mathrm{a^{-1}}} * \mathrm{mol\ m^{-3}} * \ln(-)$

Note: 1. Symbols used above and in Appendix A:

GYPBLK = g_o

DUNSAT = λ_w

AEQBM = a

TFINAL = t_f

CSINIT = C_s^o = 2R

NPRCP = P

KRATE = k

2. Original equation incorrectly uses h instead of λ_w. Corrected after discussions with SENES on 10/2/86.

Figure 1.2.3
Calculate Mixing
Zone Concentrations

March 7, 1986

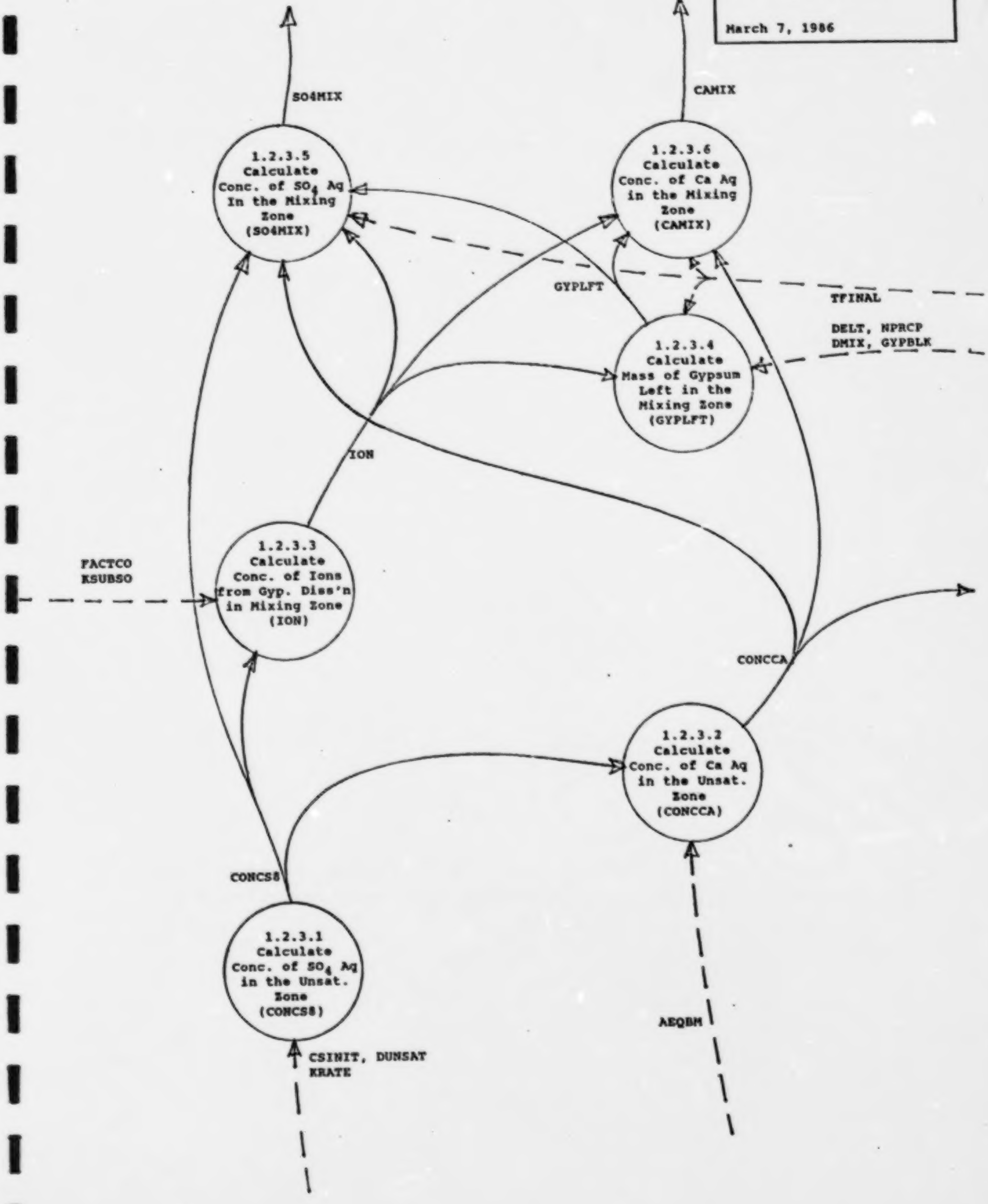
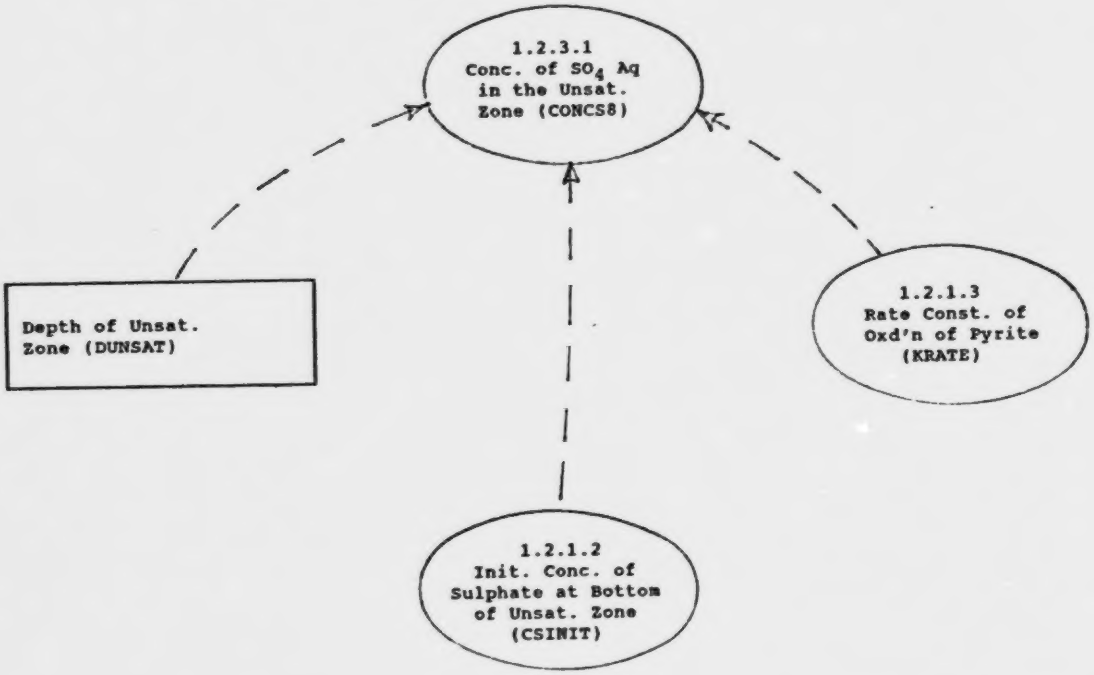

SO4MIX
CAMIX
1.2.3.5
Calculate
Conc. of SO_4 Aq
In the Mixing
Zone
(SO4MIX)
1.2.3.6
Calculate
Conc. of Ca Aq
in the Mixing
Zone
(CAMIX)
GYPLFT
TFINAL
DELT, NPRCP
DMIX, GYPBLK
1.2.3.4
Calculate
Mass of Gypsum
Left in the
Mixing Zone
(GYPLFT)
ION
FACTCO
KSUBSO
1.2.3.3
Calculate
Conc. of Ions
from Gyp. Diss'n
in Mixing Zone
(ION)
CONCCA
1.2.3.2
Calculate
Conc. of Ca Aq
in the Unsat.
Zone
(CONCCA)
CONCS8
1.2.3.1
Calculate
Conc. of SO_4 Aq
in the Unsat.
Zone
(CONCS8)
CSINIT, DUNSAT
KRATE
AEQBM

Figure 1.2.3.1
Calculate Conc. of
SO_4 Aq. in the
Unsat. Zone
(CONCS8)
March 7, 1986

CONCS8 CONC. OF SO4 AQ IN THE UNSAT. ZONE [mol m^{-3}]. Concentration of sulphate ions in the aqueous phase of the unsaturated zone, due to pyrite oxidation and downward movement of water, as a function of time and position.


1.2.3.1
Conc. of SO4 Aq
in the Unsat.
Zone (CONCS8)
Depth of Unsat.
Zone (DUNSAT)
1.2.1.3
Rate Const. of
Oxd'n of Pyrite
(KRATE)
1.2.1.2
Init. Conc. of
Sulphate at Bottom
of Unsat. Zone
(CSINIT)

$$\mathrm{CONCS8} = \frac{\mathrm{CSINIT} * \mathrm{Z}}{\mathrm{DUNSAT}} * \mathrm{EXP}\ (-\mathrm{KRATE} * \mathrm{TIME}) \qquad [\mathrm{mol\ m}^{-3}] \qquad \text{(Eq. A.3.16)}$$

Unit check: mol m^{-3} * m / m * EXP(a^{-1} * a) = [mol m^{-3}]

Notes: 1. Most, if not all, uses of CONCS8 appear to be at Z = DUNSAT, with a corresponding simplification possible. The only known exception is where PHUNST is required at Z ≠ DUNSAT. PHUNST is not used anywhere, however.

2. Symbols used in Appendix A:

CSINIT = $C_s^o = C_s(o,\lambda_w)$

CONCS8 = $C_s(t,\lambda_w) = C(t,z) = C_s = [SO_4{}^{2-}]_p = [SO_4{}^{2-}]_{\lambda w}$

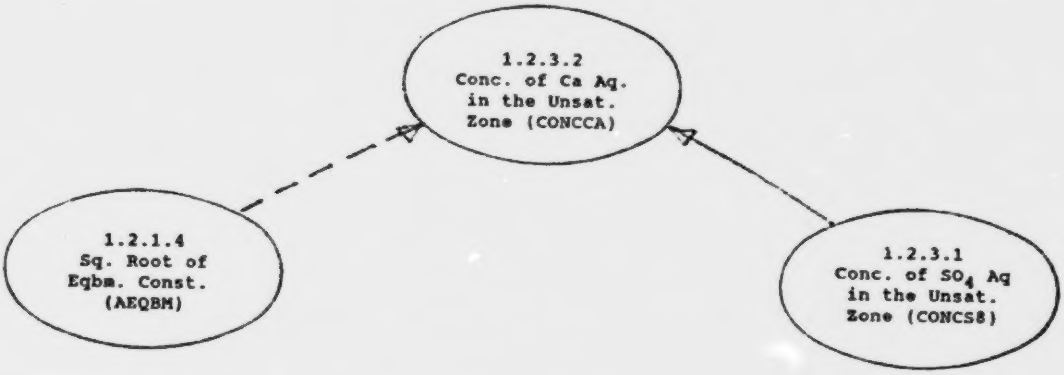
DUNSAT = λ_w

KRATE = k

Figure 1.2.3.2
Calculate Conc. of
Ca Aq. in the Unsat.
Zone (CONCCA)

March 7, 1986

CONCCA CONC. OF CA AQ. IN THE UNSAT. ZONE [mol m^{-3}]. Concentration of calcium ions in the aqueous phase of the unsaturated zone as a function of time and position.


1.2.3.2
Conc. of Ca Aq.
in the Unsat.
Zone (CONCCA)
1.2.1.4
Sq. Root of
Eqbm. Const.
(AEQBM)
1.2.3.1
Conc. of SO_4 Aq
in the Unsat.
Zone (CONCS8)

$$\mathrm{CONCCA} = \left\{ (\mathrm{AEQBM})^2 + \left(\frac{\mathrm{CONCS8}}{2}\right)^2 \right\}^{\frac{1}{2}} - \frac{\mathrm{CONCS8}}{2} \qquad [\mathrm{mol\ m^{-3}}] \qquad \text{(Eq. A.3.9 modified for clarity)}$$

Unit check: $[\mathrm{mol}^2\ \mathrm{m}^{-6}]^{\frac{1}{2}} = [\mathrm{mol\ m}^{-3}]$

Notes:

1. CONCCA $\equiv C_T \equiv C_T(t,Z)$. The time and position dependence appear in CONCS8. Note, however, that all uses of CONCCA are at Z = DUNSAT (for FRACRA, RAMIX, CAMIX, SO4MIX). PHUNST may require CONCCA as a function of Z, but PHUNST is not required anywhere.
2. The time integral of CONCCA is used in DEPTH1, TFINAL and REPGYP, but is evaluated separately in these variable definitions.
3. Symbols used in Appendix A:

   CONCCA $= [Ca^{2+}] = C_T = C_T(t,Z)$

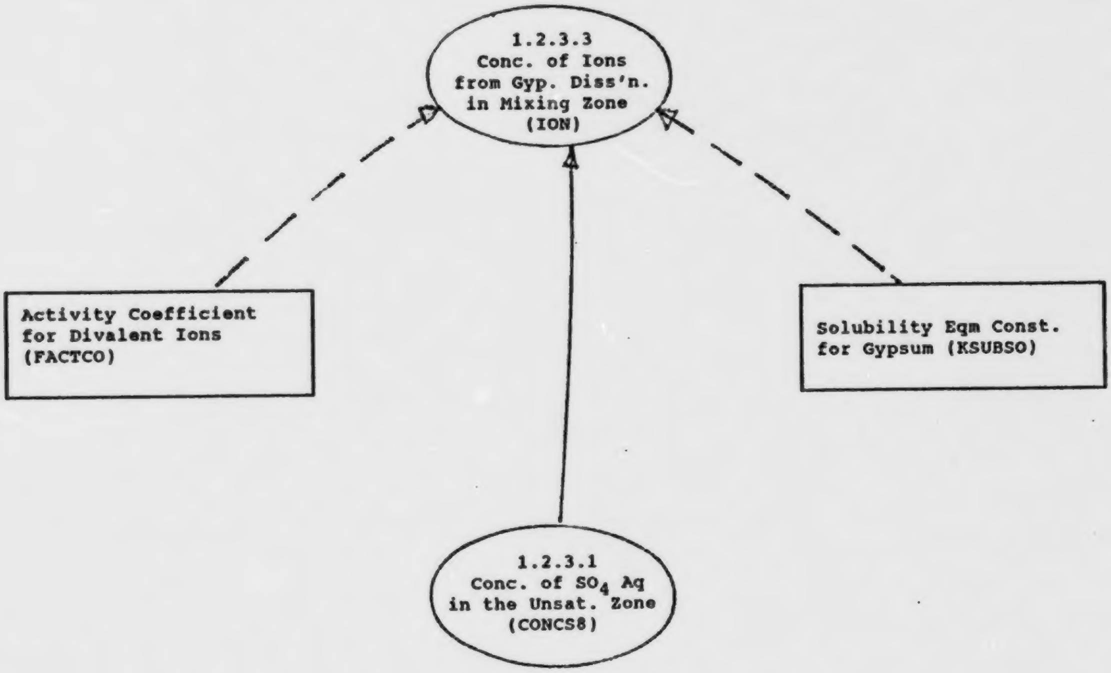
   AEQBM $= a$

   CONCS8 $= C_s(t,\lambda_w) = [SO_4^{\ 2-}]_p = [SO_4^{\ 2-}]_{\lambda w}$

Figure 1.2.3.3
Calculate Conc. of Ions
from Gyp. Diss'n. in
Mixing Zone (ION)

April 11, 1986

ION CONC. OF IONS FROM GYP. DISS'N IN MIXING ZONE [mol m^{-3}]. Concentration of Ca and SO_4 in the aqueous phase of the mixing zone, due to dissolution of gypsum.


1.2.3.3
Conc. of Ions
from Gyp. Diss'n.
in Mixing Zone
(ION)
Activity Coefficient
for Divalent Ions
(FACTCO)
Solubility Eqm Const.
for Gypsum (KSUBSO)
1.2.3.1
Conc. of SO4 Aq
in the Unsat. Zone
(CONCS8)

ION = 0 (TIME ≤ TFINAL) (no eqn.)

$$\text{ION} = -\frac{\text{CONCS8}}{2} + \frac{1}{2}\left\{ (\text{CONCS8})^2 + \frac{4 * \text{KSUBSO}}{(\text{FACTCO})^2} \right\}^{½} \quad [\text{mol. } m^{-3}] \qquad (\text{Eq. A.7.2})$$

Unit check: $\text{mol } m^{-3} + [\text{mol}^2\ m^{-6} + \text{mol}^2\ m^{-6} / (-)]^{½} = [\text{mol } m^{-3}]$

Notes:
1. CONCS8 required at Z = DUNSAT.
2. ION is a function of TIME through its dependence on CONCS8.
3. Symbols used in Appendix A:

ION = C

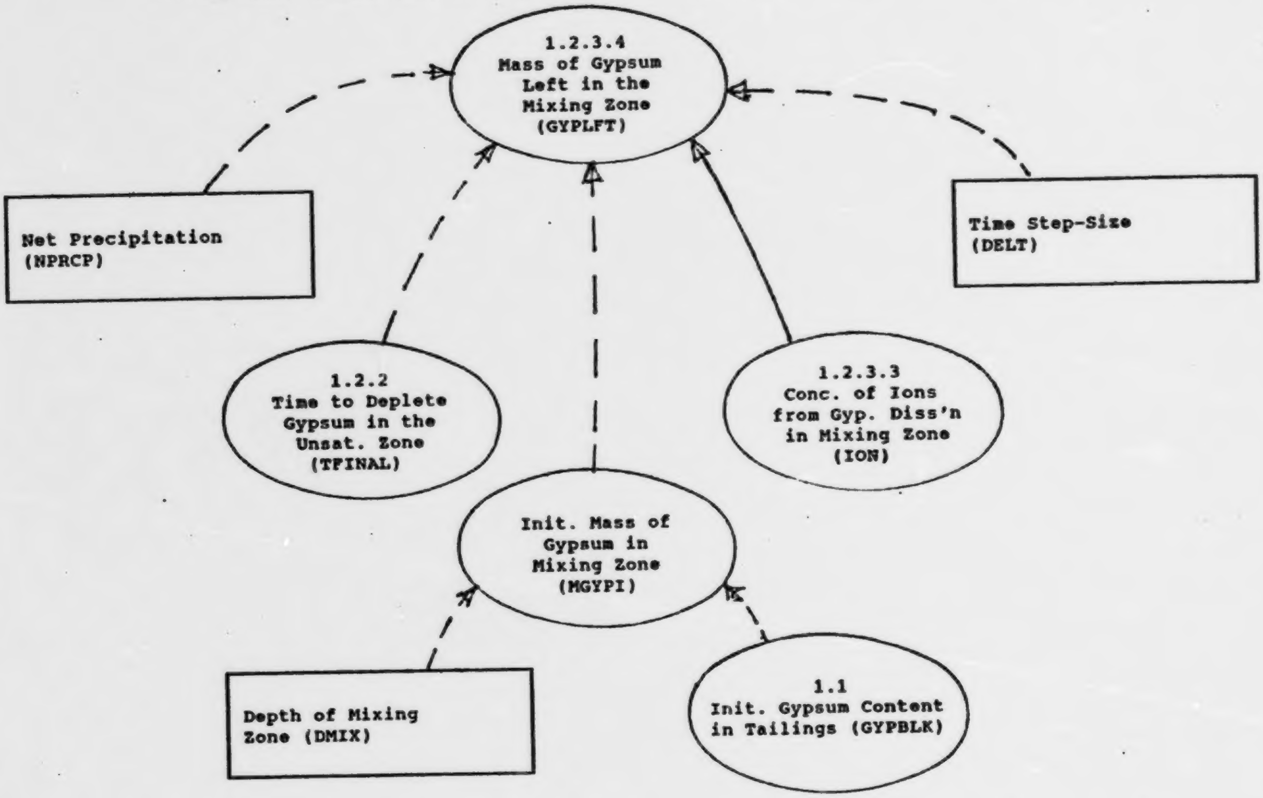
CONCS8 = $c_s(t,\lambda_w) = [SO_4{}^{2-}]_p = [SO_4{}^{2-}]_{\lambda w}$

KSUBSO = $K^o_{s,CaSO_4}$

FACTCO = f

Figure 1.2.3.4
Calculate Mass of
Gypsum Left in the
Mixing Zone
(GYPLFT)
March 7, 1986

GYPLFT MASS OF GYPSUM LEFT IN THE MIXING ZONE [mol m^{-2}]. Mass of gypsum per unit area left in the mixing zone, taking into account the dissolution that begins at TIME = TFINAL.


1.2.3.4
Mass of Gypsum
Left in the
Mixing Zone
(GYPLFT)
Net Precipitation
(NPRCP)
Time Step-Size
(DELT)
1.2.2
Time to Deplete
Gypsum in the
Unsat. Zone
(TFINAL)
1.2.3.3
Conc. of Ions
from Gyp. Diss'n
in Mixing Zone
(ION)
Init. Mass of
Gypsum in
Mixing Zone
(MGYPI)
Depth of Mixing
Zone (DMIX)
1.1
Init. Gypsum Content
in Tailings (GYPBLK)

$$\text{MGYPI} = \text{GYPBLK} * \text{DMIX} \qquad [\text{mol m}^{-2}] \qquad (\text{Eq. A.7.3})$$

Solve the following finite difference equation for GYPLFT (TIME + DELT):

for TIME $\leq$ TFINAL:

$$\text{GYPLFT (TIME)} = \text{MGYPI} \qquad [\text{mol m}^{-2}]$$

for TIME > TFINAL:

$$\text{GYPLFT (TIME + DELT)} = \text{GYPLFT (TIME)} - \text{NPRCP} * \text{DELT} * \frac{[\text{ION (TIME + DELT)} + \text{ION (TIME)}]}{2} \qquad [\text{mol m}^{-2}] \qquad (\text{from Eq. A.7.4})$$

Unit check: MGYPI $= \text{mol m}^{-3} * \text{m} = [\text{mol m}^{-2}]$

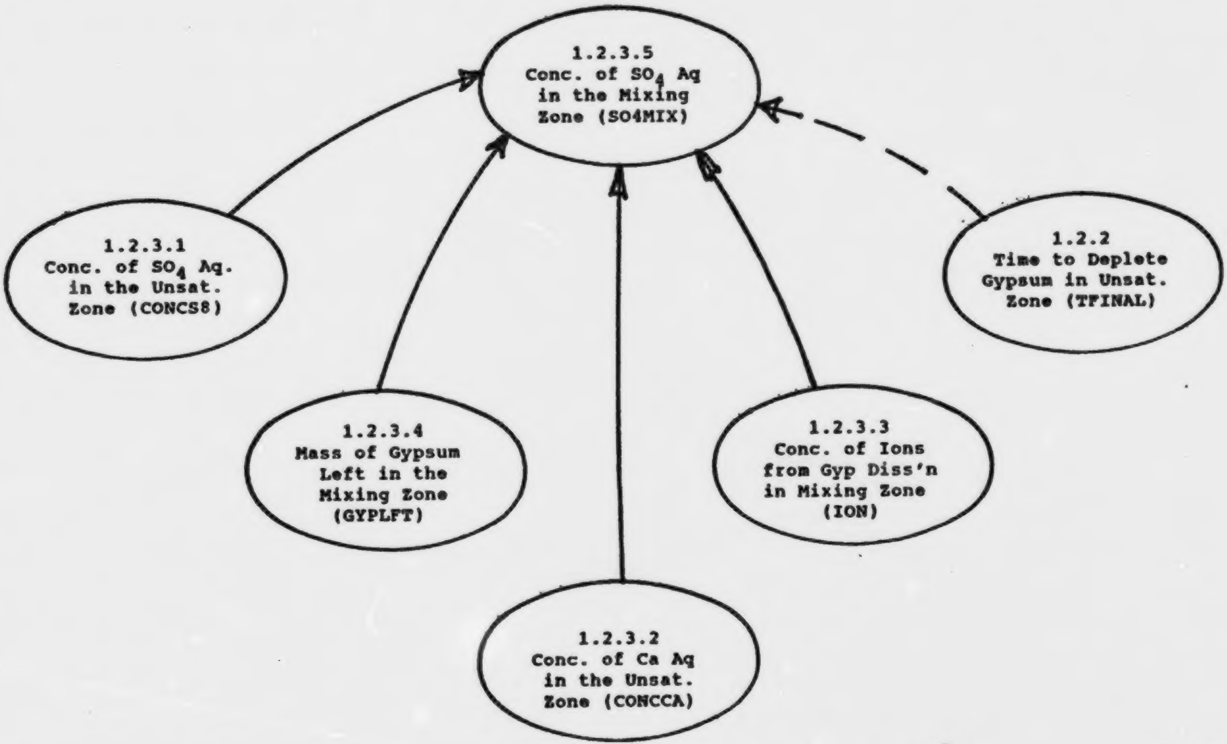
GYPLFT $= \text{mol m}^{-2} - \text{m a}^{-1} * \text{a} * \text{mol m}^{-3} = [\text{mol m}^{-2}]$

Notes:
1. Original equations have been modified to calculate the amount left, not the amount dissolved.
2. Symbols used in Appendix A:

   TFINAL = t_f
   MGYPI = M^o_{GYP}
   ION = C_n
   NPRCP = P
   DELT = Δt

Figure 1.2.3.5
Calculate Conc. of
SO_4 Aq. in the
Mixing Zone
(SO4MIX)
April 11, 1986

SO4MIX CONC. OF SO4 AQ IN THE MIXING ZONE [mol m^{-3}]. Concentration of SO_4 in the aqueous phase of the mixing zone.


1.2.3.5
Conc. of SO_4 Aq
in the Mixing
Zone (SO4MIX)
1.2.3.1
Conc. of SO_4 Aq.
in the Unsat.
Zone (CONCS8)
1.2.2
Time to Deplete
Gypsum in Unsat.
Zone (TFINAL)
1.2.3.4
Mass of Gypsum
Left in the
Mixing Zone
(GYPLFT)
1.2.3.3
Conc. of Ions
from Gyp Diss'n
in Mixing Zone
(ION)
1.2.3.2
Conc. of Ca Aq
in the Unsat.
Zone (CONCCA)

SO4MIX = CONCS8 + CONCCA	(TIME $\leq$ TFINAL)	[mol m^{-3}]	(See note 3)
SO4MIX = CONCS8 + ION	(TIME > TFINAL and GYPLFT > 0)	[mol m^{-3}]	(p. A-26)
SO4MIX = CONCS8	(TIME > TFINAL and GYPLFT $\leq$ 0)	[mol m^{-3}]	(Eq. A.7.5)

Unit check: [mol m^{-3}]

Notes:
1. CONCS8 required at Z = DUNSAT.
2. Symbols used in Appendix A:

   ION = C = C_n = $[SO_4^{2-}]_{diss}$ = $[Ca^{2+}]_{diss}$

   TFINAL = t_f

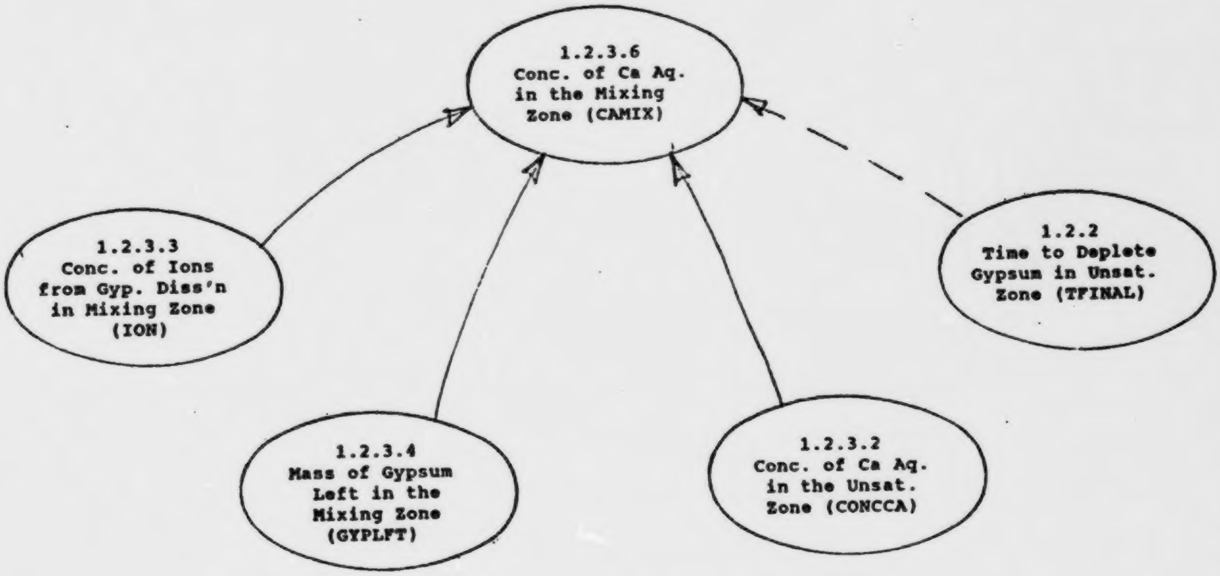
   SO4MIX = $[SO_4^{2-}]_{mix}$

   CONCS8 = $[SO_4^{2-}]_{\lambda_w} = C_s(t,\lambda_w)$
3. In equation (A.7.1), $[SO_4^{2-}]_{MIX}$ is set equal to $[SO_4^{2-}]_{\lambda w}$, but the latter is not defined elsewhere. It is assumed, following equation (A.3.8) that $[SO_4^{2-}]_{\lambda w}$ is the sum of concentrations from pyrite oxidation (CONCS8) and from gypsum dissolution (CONCCA).

Figure 1.2.3.6
Calculate Conc. of
Ca Aq. in the
Mixing Zone (CAMIX)

March 7, 1986

CAMIX CONC. OF CA AQ IN THE MIXING ZONE [mol m^{-3}]. Concentration of Ca in the aqueous phase of the mixing zone.


1.2.3.6
Conc. of Ca Aq.
in the Mixing
Zone (CAMIX)
1.2.3.3
Conc. of Ions
from Gyp. Diss'n
in Mixing Zone
(ION)
1.2.2
Time to Deplete
Gypsum in Unsat.
Zone (TFINAL)
1.2.3.4
Mass of Gypsum
Left in the
Mixing Zone
(GYPLFT)
1.2.3.2
Conc. of Ca Aq.
in the Unsat.
Zone (CONCCA)

CAMIX = CONCCA	(TIME $\leq$ TFINAL)	[mol m^{-3}]	(Eq. A.7.1)
CAMIX = ION	(TIME > TFINAL and GYPLFT > 0)	[mol m^{-3}]	(p. A-26)
CAMIX = 0	(TIME > TFINAL and GYPLFT $\leq$ 0)	[mol m^{-3}]	(Eq. A.7.5)

Unit check: [mol m^{-3}]

Notes: 1. CONCCA required at position Z = DUNSAT (see CONCCA)

2. Symbols used in Appendix A:

CAMIX = $[CA^{2+}]_{mix}$

CONCCA = $[Ca^{2+}]_{\lambda} = [Ca^{2+}]$

ION = $C = C_n^w = [SO_4{}^{2-}]_{diss} = [Ca^{2+}]_{diss}$

TFINAL = t_f

Figure 1.2.4
Calculate Position
of Gypsum Depletion
Front

March 7, 1986

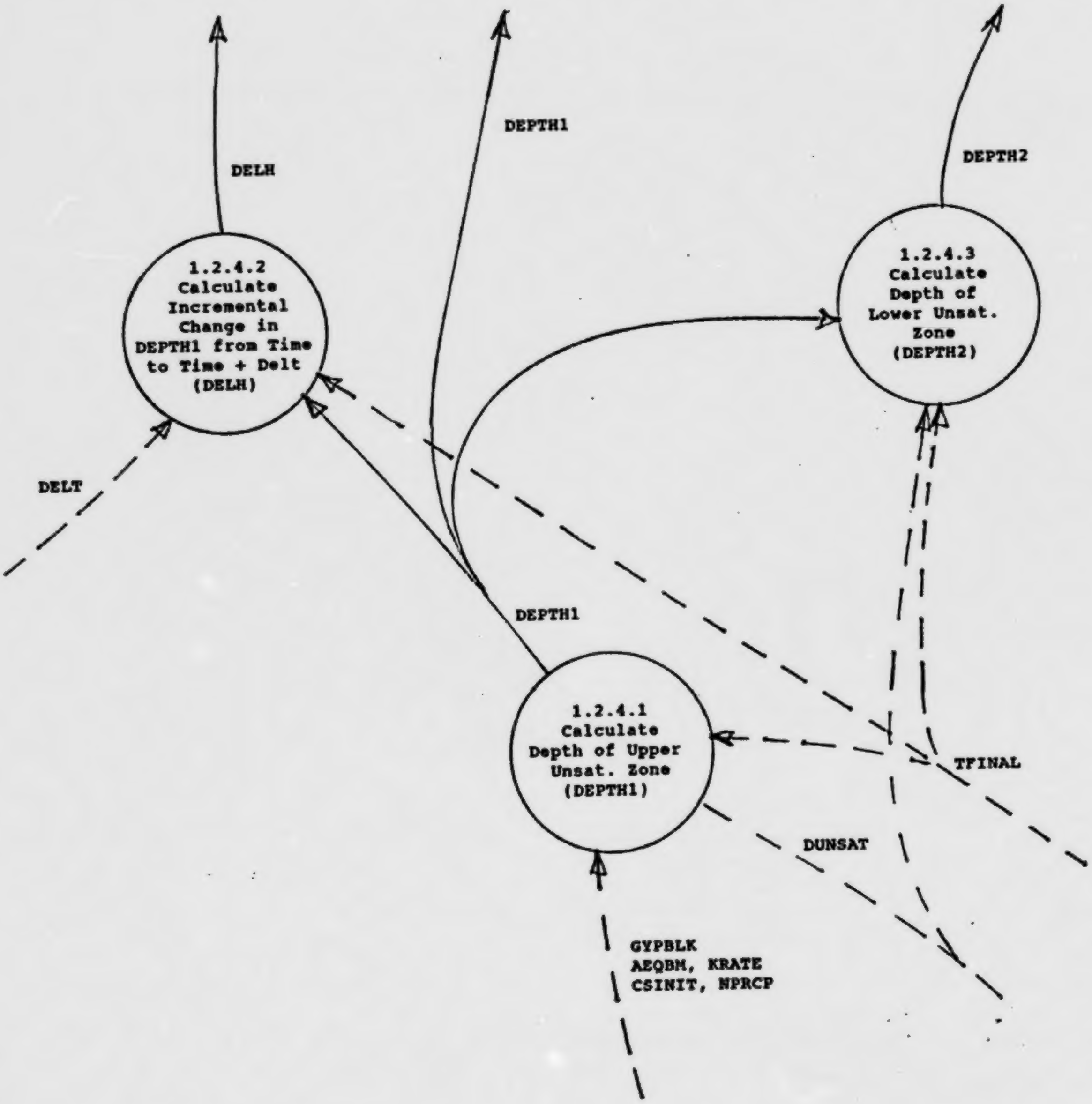
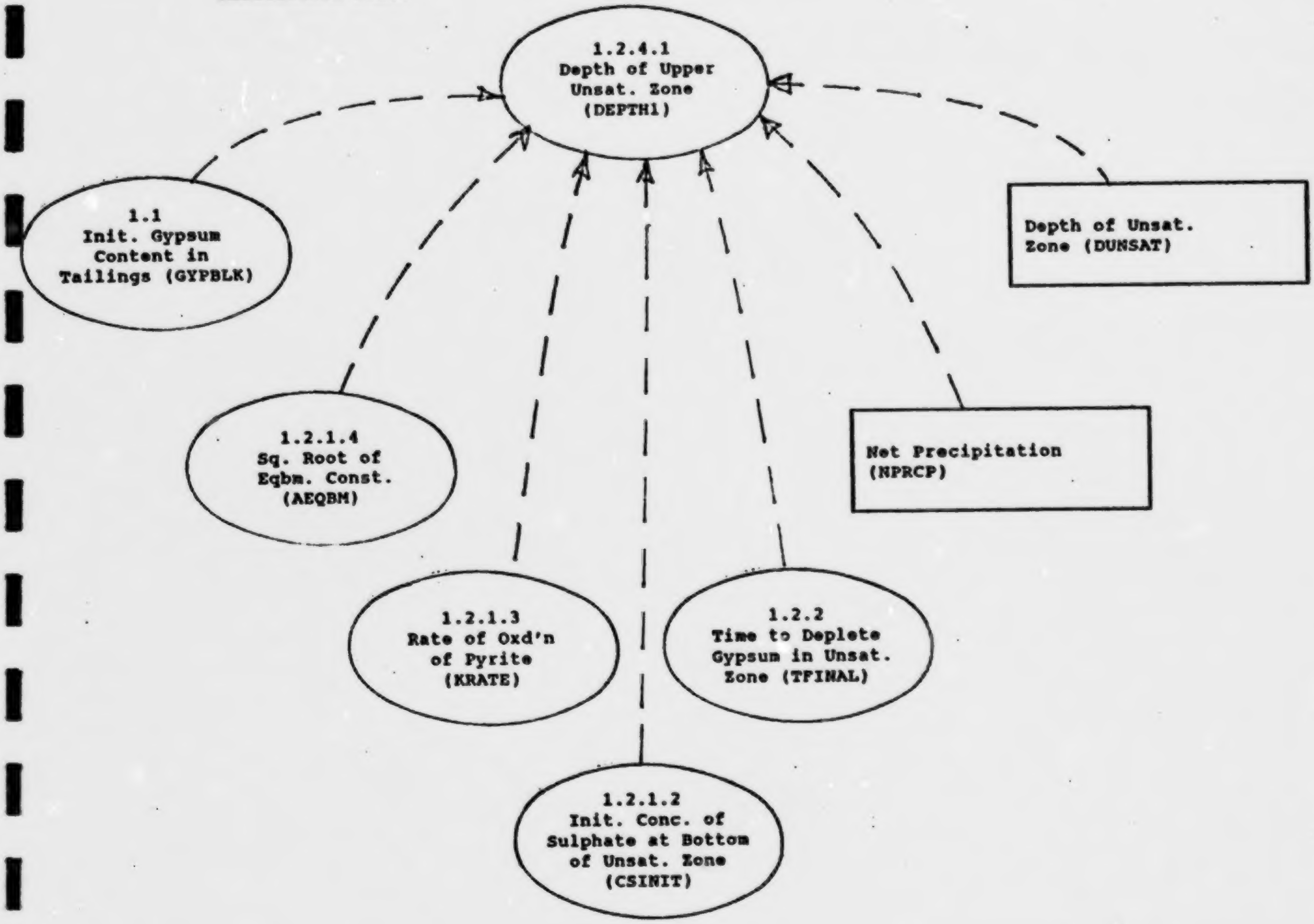

DELH
DEPTH1
DEPTH2
1.2.4.2
Calculate
Incremental
Change in
DEPTH1 from Time
to Time + Delt
(DELH)
1.2.4.3
Calculate
Depth of
Lower Unsat.
Zone
(DEPTH2)
DELT
DEPTH1
1.2.4.1
Calculate
Depth of Upper
Unsat. Zone
(DEPTH1)
TFINAL
DUNSAT
GYPBLK
AEQBM, KRATE
CSINIT, NPRCP

Figure 1.2.4.1
Calculate Depth of
Upper Unsat. Zone
(DEPTH1)

March 13, 1986

DEPTH1 DEPTH OF SUPPER UNSAT. ZONE [m]. Depth of the upper (gypsum-depleted) layer of the unsaturated zone.


1.2.4.1
Depth of Upper Unsat. Zone (DEPTH1)
1.1
Init. Gypsum Content in Tailings (GYPBLK)
Depth of Unsat. Zone (DUNSAT)
1.2.1.4
Sq. Root of Eqbm. Const. (AEQBM)
Net Precipitation (NPRCP)
1.2.1.3
Rate of Oxd'n of Pyrite (KRATE)
1.2.2
Time to Deplete Gypsum in Unsat. Zone (TFINAL)
1.2.1.2
Init. Conc. of Sulphate at Bottom of Unsat. Zone (CSINIT)

DEPTH1 = DUNSAT (TIME ≥ TFINAL) [m] (no Eqn)

for TIME < TFINAL, solve the following equation iteratively for DEPTH1

$$g_o h = Pat + \frac{P}{k}\left\{(a^2 + S^2)^{1/2} - \left(a^2 + S^2 e^{-2kt}\right)^{1/2}\right\} - \frac{PS}{k}(1 - e^{-kt}) - \frac{Pa}{k}\ln\left\{\frac{a + (a^2 + S^2)^{1/2}}{a + \left(a^2 + S^2 e^{-2kt}\right)^{1/2}}\right\} \quad \text{(Eq. A.3.20)}$$

Initial guess: $h - \frac{\text{DUNSAT}}{2}$

Unit check: $\text{mol m}^{-3} * \text{m} = \text{m a}^{-1} * \text{mol m}^{-3} * \text{a} + \text{m a}^{-1} / [\text{a}^{-1}] * \text{mol m}^{-3} - \text{m a}^{-1} * \text{mol m}^{-3} / [\text{a}^{-1}]$
$- \text{m a}^{-1} * \text{mol m}^{-3} / [\text{a}^{-1}] * \ln[-] = [\text{mol m}^{-2}]$

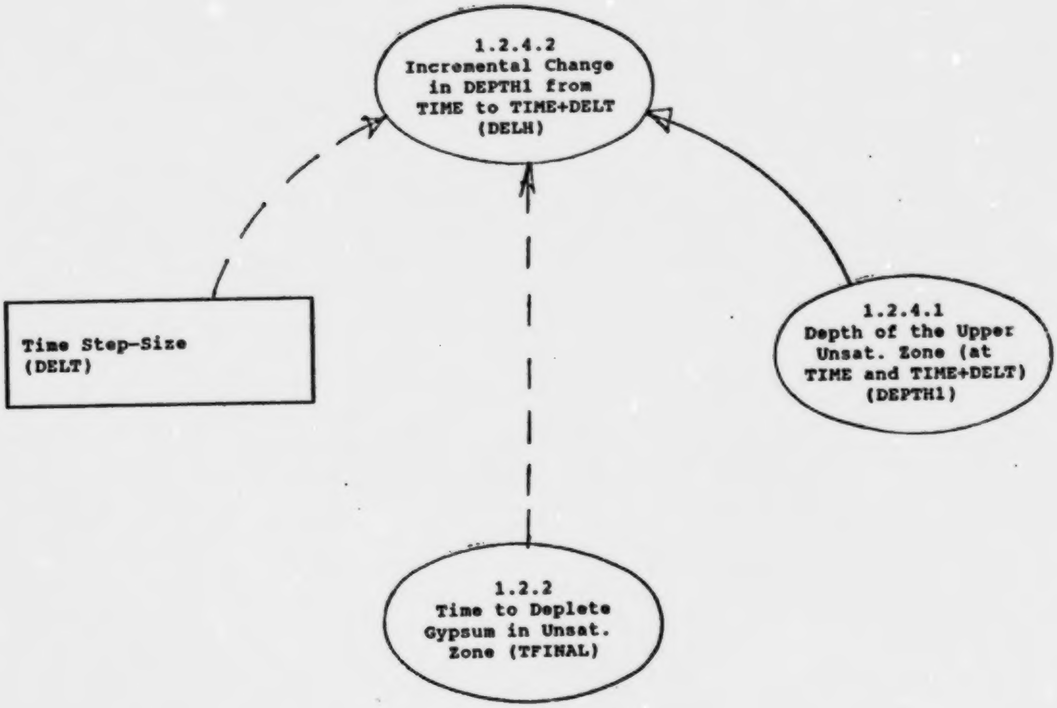
Note: Symbols used above and in Appendix A:

GYPBLK = g_o
DEPTH1 = h
NPRCP = P
DUNSAT = λ_w
CSINIT = c_s^o

KRATE = k
TIME = t
AEQBM = a
$S = c_s^o * h / (2\lambda_w)$ [mol m^{-3}]
]

Figure 1.2.4.2
Calculate Incremental
Change in DEPTH1 from
TIME to TIME + DELT
(DELH)
March 12, 1986

DELH INCREMENTAL CHANGE IN DEPTH1 FROM TIME TO TIME+DELT [m]. Incremental change in the depth of the upper (gypsum-depleted) layer of the unsaturated zone from time = TIME to time = TIME+DELT.


1.2.4.2
Incremental Change in DEPTH1 from TIME to TIME+DELT (DELH)
Time Step-Size (DELT)
1.2.4.1
Depth of the Upper Unsat. Zone (at TIME and TIME+DELT) (DEPTH1)
1.2.2
Time to Deplete Gypsum in Unsat. Zone (TFINAL)

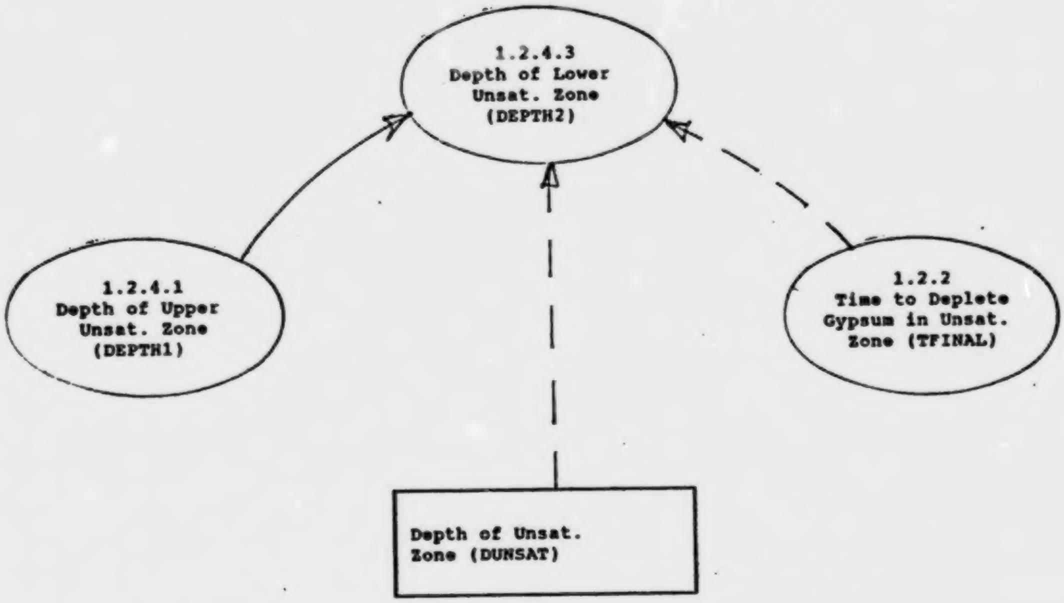
DELH = DEPTH1 (TIME+DELT) - DEPTH1 (TIME)	TIME < TFINAL	[m]	(B. Goodwin)
DELH = 0	TIME ≥ TFINAL	[m]	(B. Goodwin)

Unit check: [m]

Figure 1.2.4.3
Calculate Depth of
Lower Unsat. Zone
(DEPTH2)

March 12, 1986

DEPTH2 DEPTH OF LOWER UNSAT. ZONE [m]. Depth of the lower (gypsum-enriched) layer of the unsaturated zone.

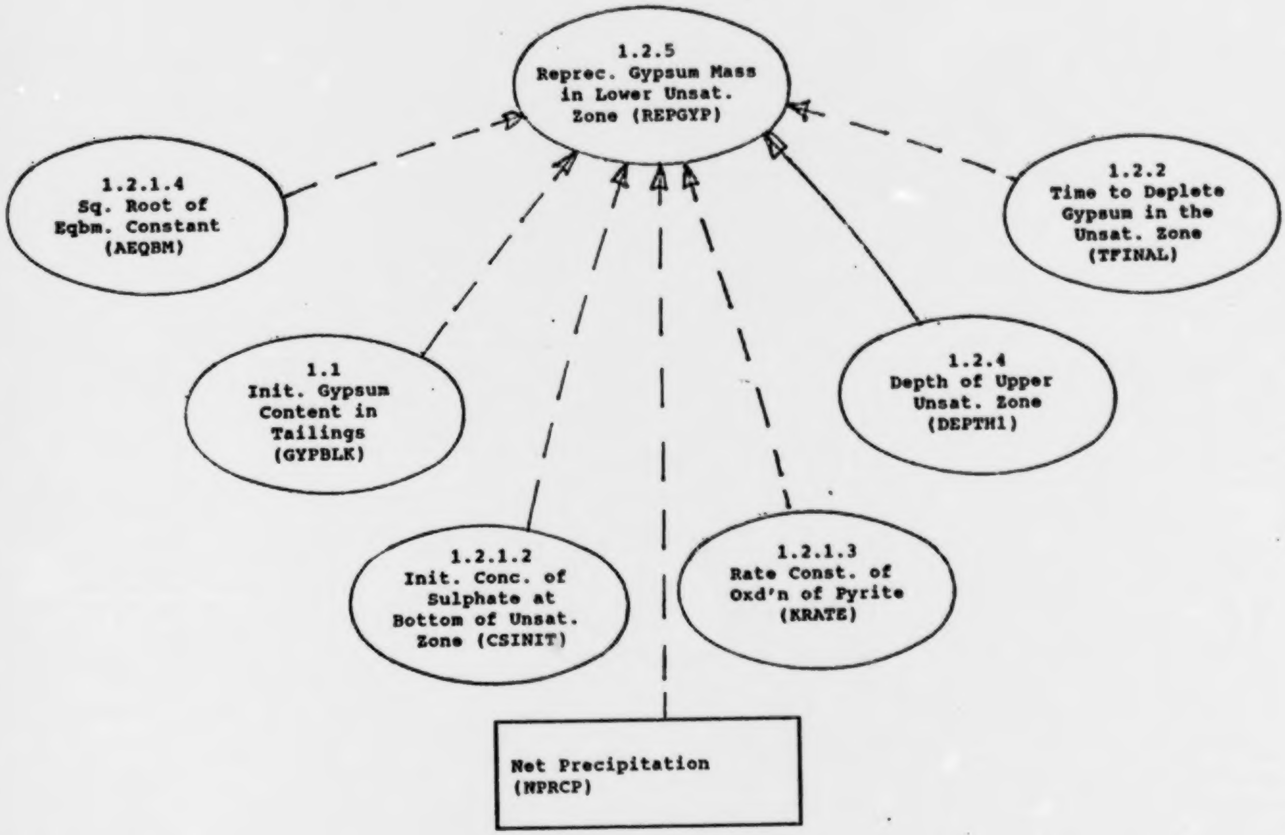

1.2.4.3
Depth of Lower Unsat. Zone (DEPTH2)
1.2.4.1
Depth of Upper Unsat. Zone (DEPTH1)
1.2.2
Time to Deplete Gypsum in Unsat. Zone (TFINAL)
Depth of Unsat. Zone (DUNSAT)

DEPTH2 = DUNSAT − DEPTH1	(TIME < TFINAL)	[m]	(B. Goodwin)
DEPTH2 = 0	(TIME ≥ TFINAL)	[m]	(B. Goodwin)

Unit check: [m]

Figure 1.2.5
Calculate Reprec.
Gypsum Mass in the
Lower Unsat. Zone
(REPGYP)
March 13, 1986

REPGYP REPREC. GYPSUM MASS IN LOWER UNSAT. ZONE [mol m^{-2}]. Reprecipitated gypsum content per unit area in the bottom (gypsum-enriched) layer of the unsaturated zone, as a function of time and depth.


1.2.5
Reprec. Gypsum Mass in Lower Unsat. Zone (REPGYP)
1.2.1.4
Sq. Root of Eqbm. Constant (AEQBM)
1.2.2
Time to Deplete Gypsum in the Unsat. Zone (TFINAL)
1.1
Init. Gypsum Content in Tailings (GYPBLK)
1.2.4
Depth of Upper Unsat. Zone (DEPTH1)
1.2.1.2
Init. Conc. of Sulphate at Bottom of Unsat. Zone (CSINIT)
1.2.1.3
Rate Const. of Oxd'n of Pyrite (KRATE)
Net Precipitation (NPRCP)

For TIME ≤ TFINAL:

$$\text{REPGYP} = g_o h - Pat - \frac{P}{k}\left\{(a^2 + R^2)^{1/2} - \left(a^2 + R^2 e^{-2kt}\right)^{1/2}\right\} + \frac{PR}{k}\left(1 - e^{-kt}\right) + \frac{Pa}{k}\ln\left\{\frac{a + \left(a^2 + R^2\right)^{1/2}}{a + \left(a^2 + R^2 e^{-2kt}\right)^{1/2}}\right\}$$

[mol m^{-2}] (Eq. A.3.22)

(B. Goodwin)

for TIME > TFINAL, REPGYP = 0.0

Unit check: mol m^{-3} * m - m a^{-1} * mol m^{-3} * a - m a^{-1} / [a^{-1}] * mol m^{-3} + m a^{-1} * mol m^{-3} / [a^{-1}] + m a^{-1} * mol m^{-3} / [a^{-1}] * ln[-] = [mol m^{-2}]

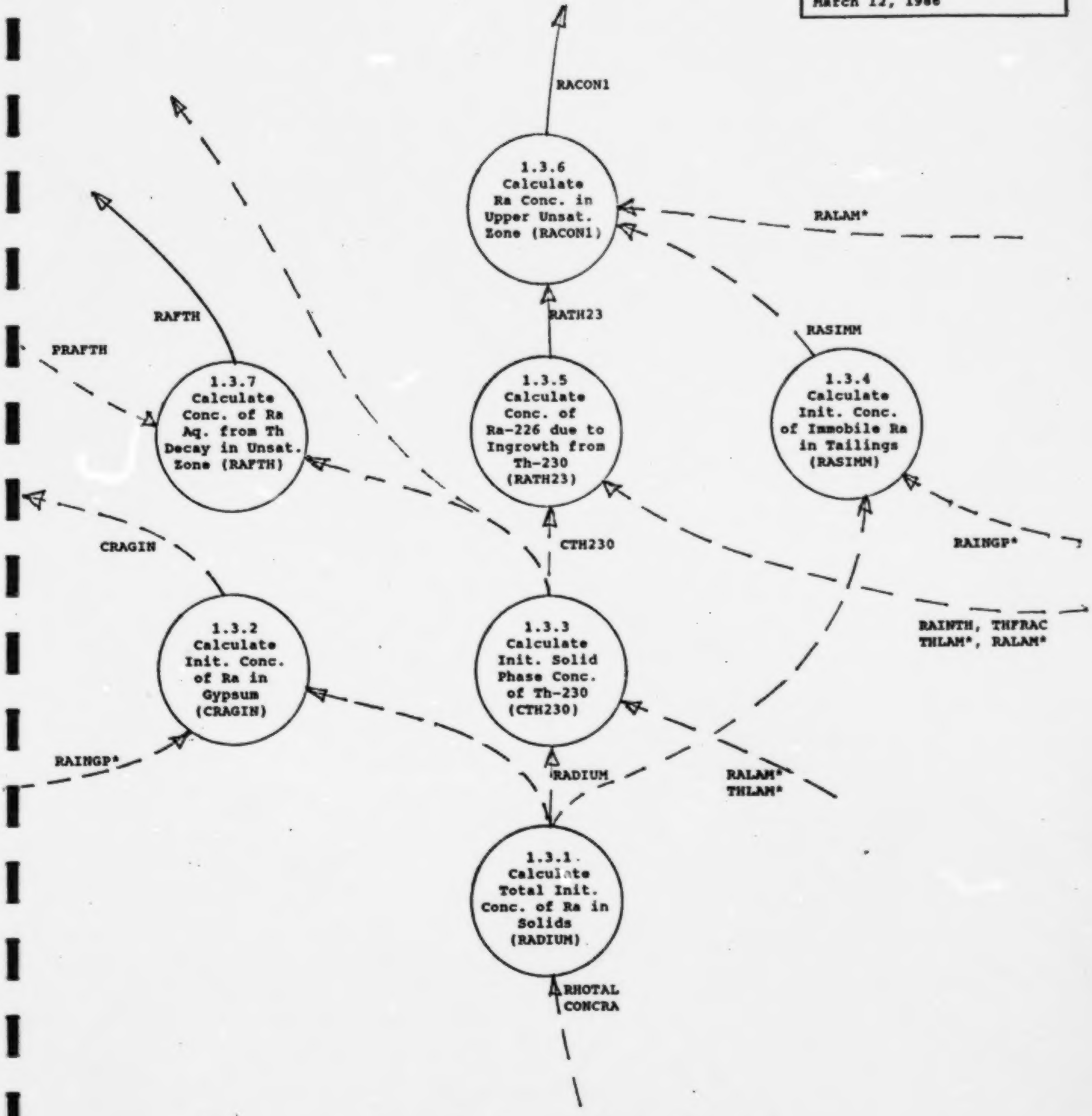
Note: Symbols used above and in Appendix A:

DUNSAT = λ_w
AEQBM = a
CSINIT = C_s^o = 2R
KRATE = k
DEPTH1 = h

TFINAL = t_f
TIME = t
NPRCP = P
GYPBLK = g_o
REPGYP = $m(t,\lambda_w - h)$

Figure 1.3
Calculate Ra Conc.
in Upper Unsat.
Zone

March 12, 1986


RACON1
1.3.6
Calculate
Ra Conc. in
Upper Unsat.
Zone (RACON1)
RALAM*
RAFTH
RATH23
RASIMM
PRAFTH
1.3.7
Calculate
Conc. of Ra
Aq. from Th
Decay in Unsat.
Zone (RAFTH)
1.3.5
Calculate
Conc. of
Ra-226 due to
Ingrowth from
Th-230
(RATH23)
1.3.4
Calculate
Init. Conc.
of Immobile Ra
in Tailings
(RASIMM)
CRAGIN
CTH230
RAINGP*
RAINTH, THFRAC
THLAM*, RALAM*
1.3.2
Calculate
Init. Conc.
of Ra in
Gypsum
(CRAGIN)
1.3.3
Calculate
Init. Solid
Phase Conc.
of Th-230
(CTH230)
RAINGP*
RADIUM
RALAM*
THLAM*
1.3.1
Calculate
Total Init.
Conc. of Ra in
Solids
(RADIUM)
RHOTAL
CONCRA

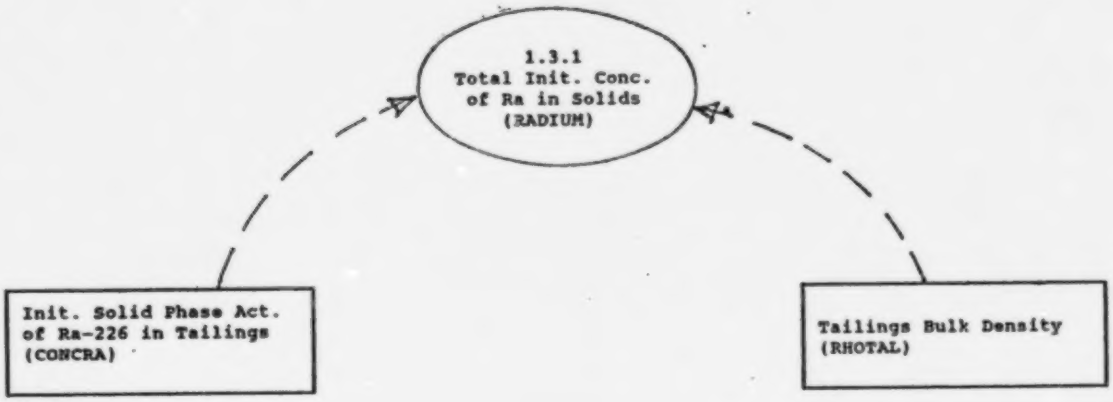
Note: PRAFTH is a cluster of input (sampled) parameters consisting of:

DUNSAT + NPRCP + RAINTH + RHOTAL + THDISS + THFRAC + THLAM + UO2TON + UPERTH.

Figure 1.3.1
Calculate Total Init.
Conc. of Ra in Solids
(RADIUM)

March 12, 1986

RADIUM TOTAL INIT. CONC. OF RA IN SOLIDS [mol m^{-3}]. Total initial solid phase concentration of Ra-226 in the tailings.


1.3.1
Total Init. Conc.
of Ra in Solids
(RADIUM)
Init. Solid Phase Act.
of Ra-226 in Tailings
(CONCRA)
Tailings Bulk Density
(RHOTAL)

RADIUM = 1.196×10^{-13} * CONCRA * RHOTAL [mol m^{-3}] (Eq. A.2.2)

Unit check: the constant 1.196×10^{-13} has units of [mol Bq^{-1}] and is a conversion factor for moles to Bequerels for Ra-226.

$\text{mol } Bq^{-1} * Bq\ g^{-1} * g\ m^{-3} = [\text{mol } m^{-3}]$

Note: Symbols used in Appendix A:

RADIUM = $[Ra_s]^o$

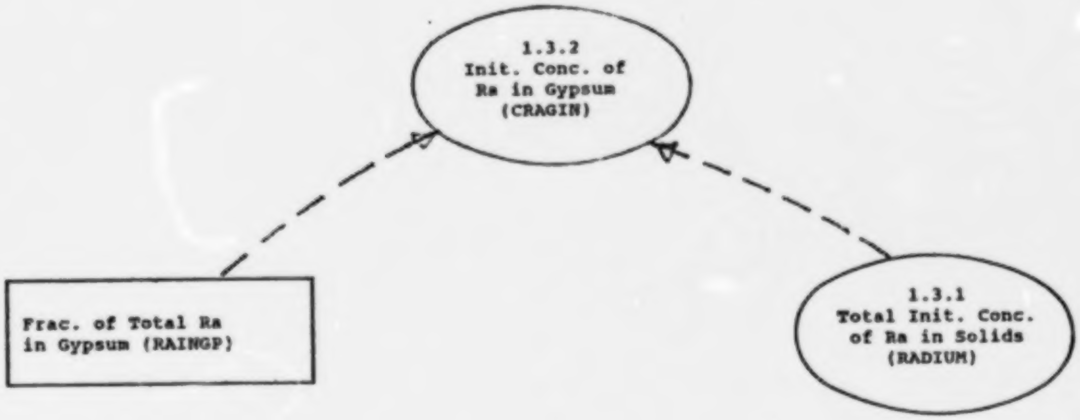
RHOTAL = ρ_T

CONCRA = Ra_s^o

Figure 1.3.2
Calculate Init.
Conc. of Ra in
Gypsum (CRAGIN)

May 12, 1986

CRAGIN INIT. CONC. OF RA IN GYPSUM [mol m^{-3}]. Initial concentration of Ra-226 associated with gypsum in the tailings.


1.3.2
Init. Conc. of
Ra in Gypsum
(CRAGIN)
Frac. of Total Ra
in Gypsum (RAINGP)
1.3.1
Total Init. Conc.
of Ra in Solids
(RADIUM)

CRAGIN = RAINGP * RADIUM [mol m^{-3}] (p. A-3)

Unit check: $[-] * \text{mol m}^{-3} = [\text{mol m}^{-3}]$

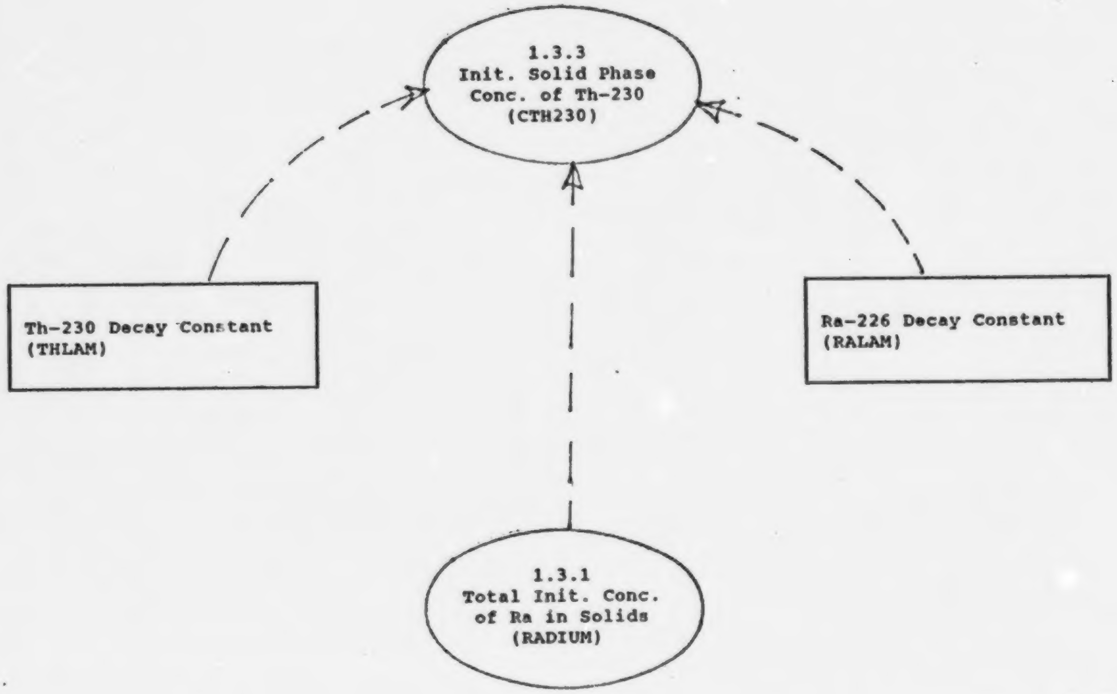
Note: Symbols used in Appendix A:

CRAGIN = $[Ra_s]^o_{gyp}$
RADIUM = $[Ra_s]^o$
RAINGP = $X_{Ra/Gyp}$

Figure 1.3.3
Calculate Init.
Solid Phase Conc.
of Th-230 (CTH230)

March 12, 1986

CTH230 INIT. SOLID PHASE CONC. OF TH-230 [mol m^{-3}]. Initial solid phase concentration of Th-230 in the tailings.


1.3.3
Init. Solid Phase
Conc. of Th-230
(CTH230)
Th-230 Decay Constant
(THLAM)
Ra-226 Decay Constant
(RALAM)
1.3.1
Total Init. Conc.
of Ra in Solids
(RADIUM)

$$CTH230 = RADIUM * \frac{RALAM}{THLAM} \quad [mol\ m^{-3}] \quad (Eq.\ A.2.3)$$

Unit check: $mol\ m^{-3} * a / a = [mol\ m^{-3}]$

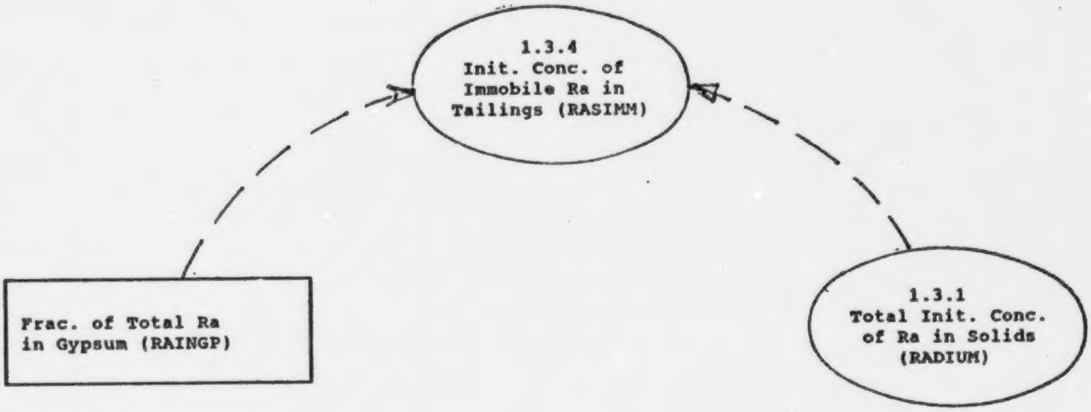
Note: Symbols used in Appendix A:

RADIUM = $[Ra_g]^o$
RALAM = λ_{Ra-226}
THLAM = λ_{Th-230}
CTH230 = $[Th\text{-}230_g]^o$ = $[Th\text{-}230]^o$

Figure 1.3.4
Calculate Init. Conc.
of Immobile Ra in
Tailings (RASIMM)

March 12, 1986

RASIMM INIT. CONC. OF IMMOBILE RA IN TAILINGS [mol m^{-3}]. Initial concentration of Ra associated with host tailings material assumed not to be leachable.


1.3.4
Init. Conc. of
Immobile Ra in
Tailings (RASIMM)
Frac. of Total Ra
in Gypsum (RAINGP)
1.3.1
Total Init. Conc.
of Ra in Solids
(RADIUM)

RASIMM = (1 - RAINGP) * RADIUM [mol m^{-3}] (p. A-3)

Unit check: [-] * mol m^{-3} = [mol m^{-3}]

Note: Symbols used in Appendix A:

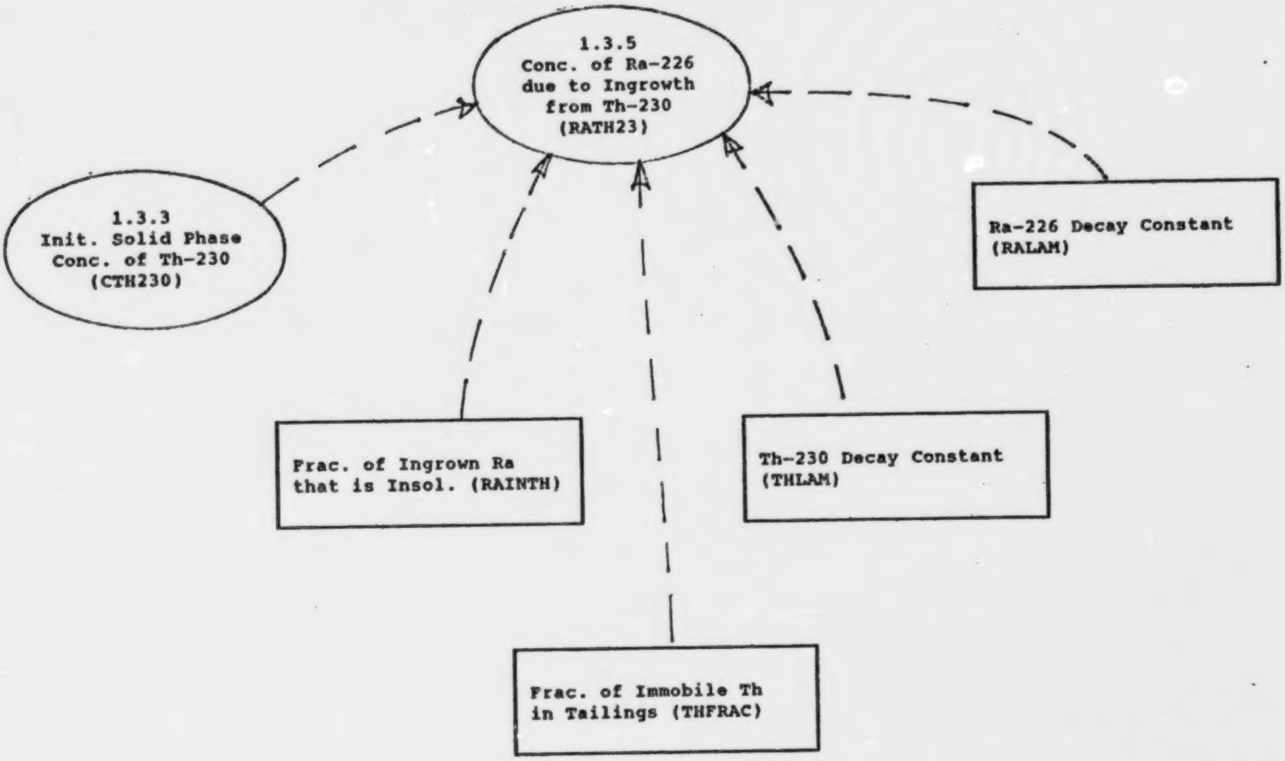
RAINGP = $x_{Ra/gyp}$

RADIUM = $[Ra_s]^{o}$

RASIMM = $[Ra_s]^{o}_{m}$

Figure 1.3.5
Calculate Conc. of
Ra-226 due to
Ingrowth from Th-230
(RATH23)
March 13, 1986

RATH23 CONC. OF RA-226 DUE TO INGROWTH FROM TH-230 [mol m^{-3}]. Concentration of Ra-226 in the solid phase due to ingrowth from the Th matrix.


1.3.5
Conc. of Ra-226
due to Ingrowth
from Th-230
(RATH23)
1.3.3
Init. Solid Phase
Conc. of Th-230
(CTH230)
Ra-226 Decay Constant
(RALAM)
Frac. of Ingrown Ra
that is Insol. (RAINTH)
Th-230 Decay Constant
(THLAM)
Frac. of Immobile Th
in Tailings (THFRAC)

$$\mathrm{RATH23} = \frac{\mathrm{RAINTH} * \mathrm{THFRAC} * \mathrm{CTH230} * \mathrm{THLAM}}{\mathrm{RALAM} - \mathrm{THLAM}} * \left\{\mathrm{EXP}(-\mathrm{THLAM} * \mathrm{TIME}) - \mathrm{EXP}(-\mathrm{RALAM} * \mathrm{TIME})\right\} .$$

[mol m^{-3}] (Eq. A.5.6)

Unit check: $(-) * (-) * \mathrm{mol}\ m^{-3} * a^{-1} / a^{-1} * \mathrm{EXP}\ (a^{-1} * a) = [\mathrm{mol}\ m^{-3}]$

Notes: 1. Symbols used in Appendix A:

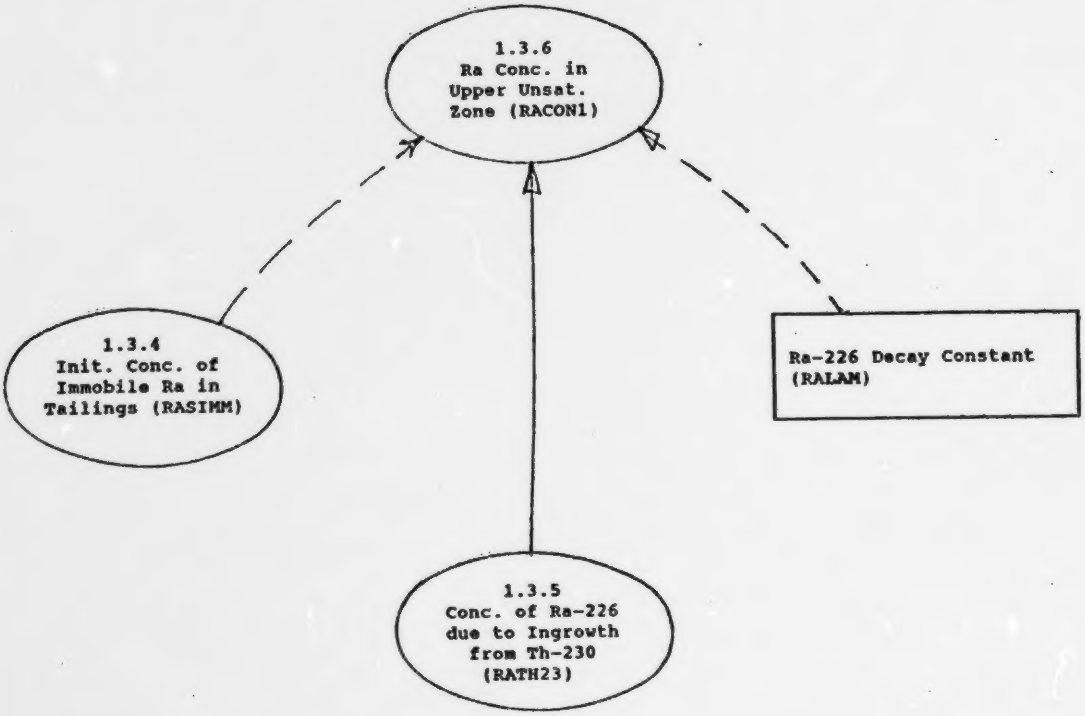
RAINTH = $x_{Ra/Th}$
THFRAC = x_{Th}
CTH230 = $[\text{Th-230}]^{\circ} \equiv [\text{Th-230}_s]^{\circ}$
THLAM = λ_{Th}

RALAM = λ_{Ra}
RATH23 = $[Ra_s]_{Th}$
TIME = t

2. $[\text{Th-230}_s]^{\circ}$ is assumed to be equivalent to $[\text{Th-230}]^{\circ}$ = CTH230 (B. Goodwin).

Figure 1.3.6
Calculate Ra
Conc. in Upper
Unsat. Zone
(RACON1)
March 13, 1986

RACON1 RA CONC. IN UPPER UNSAT. ZONE [mol m^{-3}]. Concentration of Ra-226 in the solid phase in the upper (gypsum-depleted) layer of the unsaturated zone.


1.3.6
Ra Conc. in
Upper Unsat.
Zone (RACON1)
1.3.4
Init. Conc. of
Immobile Ra in
Tailings (RASIMM)
Ra-226 Decay Constant
(RALAM)
1.3.5
Conc. of Ra-226
due to Ingrowth
from Th-230
(RATH23)

$$\text{RACON1} = \text{RASIMM} * \text{EXP}(-\text{RALAM} * \text{TIME}) + \text{RATH23} \qquad [\text{mol m}^{-3}] \qquad \text{(Eq. A.6.1)}$$

Unit checks: $\text{mol m}^{-3} * \text{EXP}(a^{-1} * a) + \text{mol m}^{-3} = [\text{mol m}^{-3}]$

Note: Symbols used in Appendix A:

RACON1 = $[Ra_s]_1$

RASIMM = $[Ra_s]_m^o$

TIME = t

RALAM = λ_{Ra}

RATH23 = $[Ra_s]_{Th}$

Figure 1.3.7
Calculate Conc. of
Ra Aq. from Th
Decay in Unsat. Zone

March 13, 1986

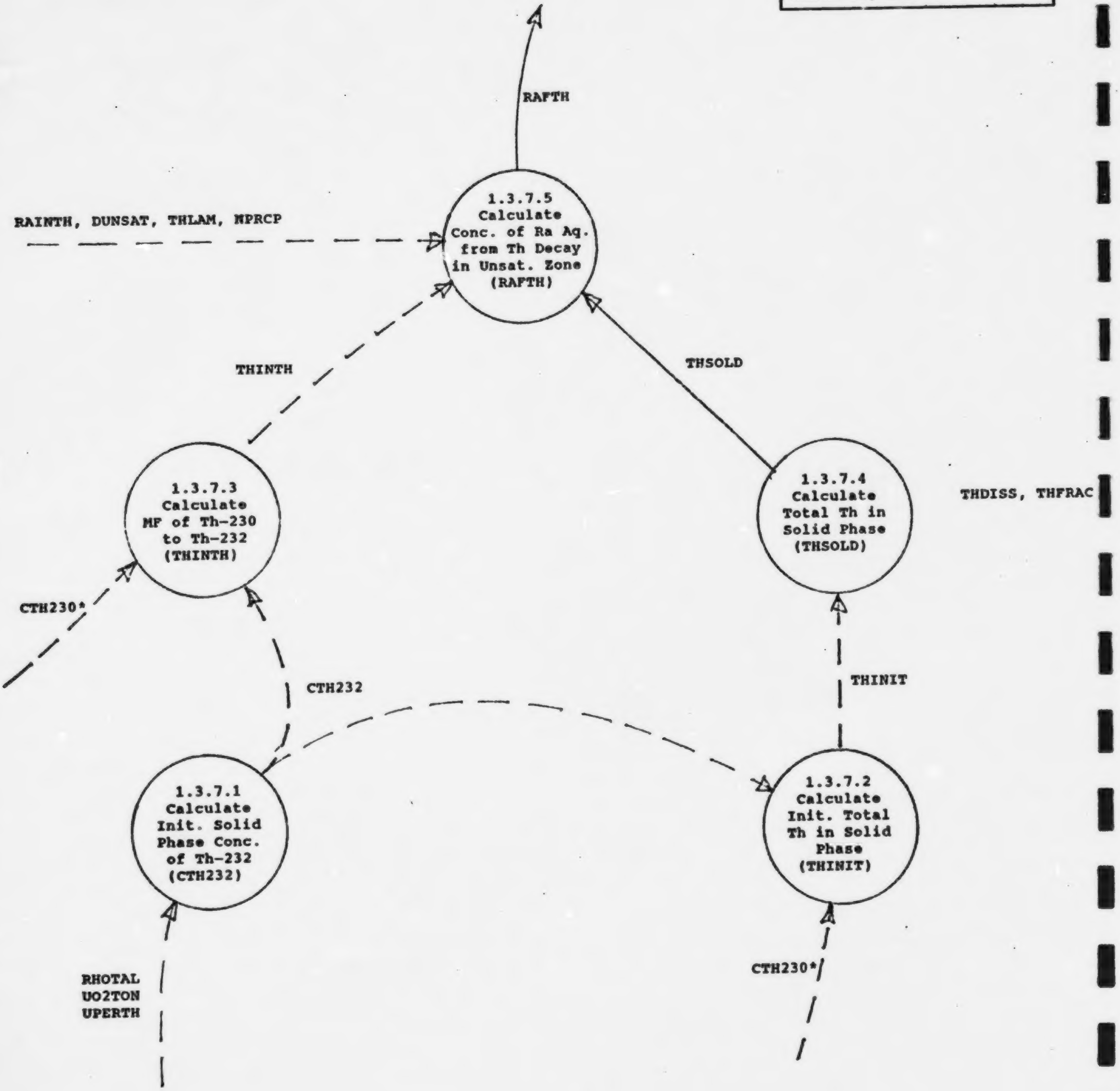
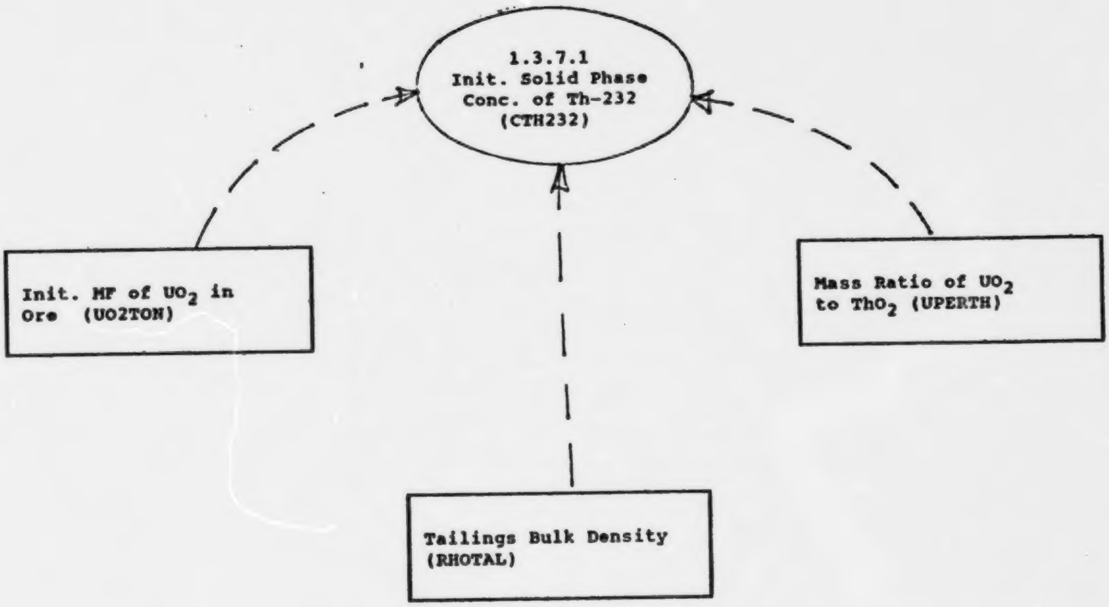

RAFTH
RAINTH, DUNSAT, THLAM, NPRCP
1.3.7.5 Calculate Conc. of Ra Aq. from Th Decay in Unsat. Zone (RAFTH)
THINTH
THSOLD
1.3.7.3 Calculate MF of Th-230 to Th-232 (THINTH)
1.3.7.4 Calculate Total Th in Solid Phase (THSOLD)
THDISS, THFRAC
CTH230*
CTH232
THINIT
1.3.7.1 Calculate Init. Solid Phase Conc. of Th-232 (CTH232)
1.3.7.2 Calculate Init. Total Th in Solid Phase (THINIT)
RHOTAL
UO2TON
UPERTH
CTH230*

Figure 1.3.7.1
Calculate Init. Solid
Phase Conc. of Th-232
(CTH232)

March 13, 1986

CTH232 INIT. SOLID PHASE CONC. OF TH-232 [mol m^{-3}]. Initial solid phase concentration of Th-232 in the tailings.


1.3.7.1
Init. Solid Phase
Conc. of Th-232
(CTH232)
Init. MF of UO_2 in Ore (UO2TON)
Mass Ratio of UO_2 to ThO_2 (UPERTH)
Tailings Bulk Density (RHOTAL)

$$CTH232 = \frac{UO2TON}{UPERTH} * RHOTAL * 3.788 \times 10^{-6} \qquad [mol\ m^{-3}] \qquad (Eq.\ A.2.4)$$

Unit check: kg $tonne^{-1}$ / (-) * g m^{-3} * mol kg^{-1} tonne g^{-1} = [mol m^{-3}]

Notes: 1. The constant has units of mol kg^{-1} tonne g^{-1} and contains factors converting moles of the ThO_2 to kg and tonnes to grams (See SENES, Vol. 3., p A-4).

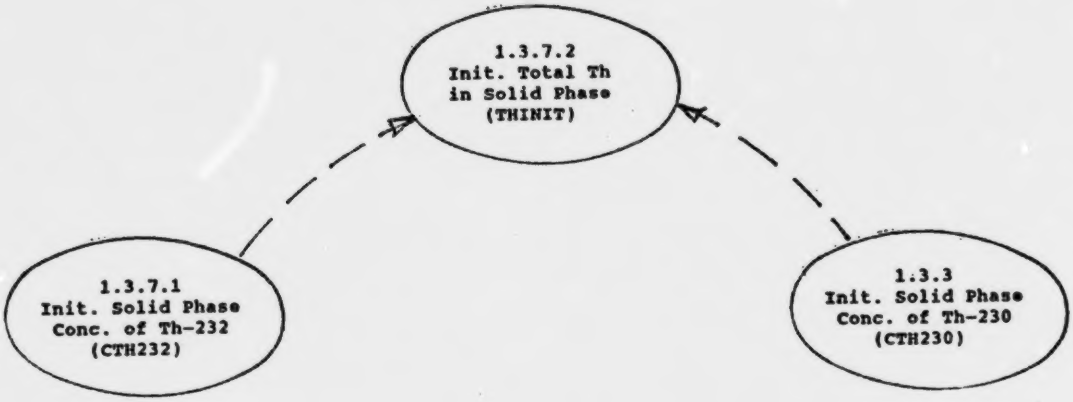
2. Symbols used in Appendix A:

CTH232 = $[TH\text{-}232_s]^o$
UO2TON = X_{UO2}
UPERTH = $X_{U/Th}$
RHOTAL = ρ_T

Figure 1.3.7.2
Calculate Init. Total
Th in Solid Phase
(THINIT)

March 13, 1986

THINIT INIT. TOTAL TH IN SOLID PHASE [mol m^{-3}]. Initial concentration of total Th in the solid phase of the unsaturated zone (cf. THSOLD).


1.3.7.2
Init. Total Th
in Solid Phase
(THINIT)
1.3.7.1
Init. Solid Phase
Conc. of Th-232
(CTH232)
1.3.3
Init. Solid Phase
Conc. of Th-230
(CTH230)

THINIT = CTH230 + CTH232 [mol m^{-3}] (Eq. A.2.5)

Unit check: [mol m^{-3}]

Note: Symbols used in Appendix A:

$THINIT = [Th_s]^\circ$

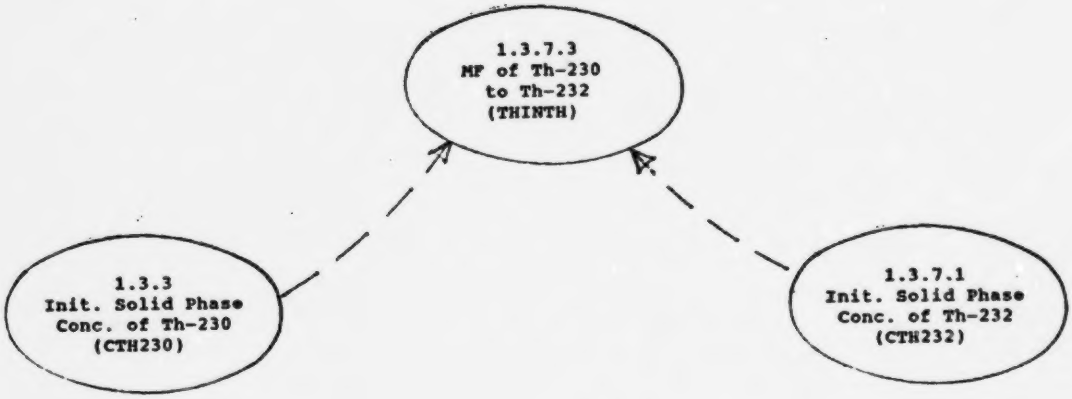
$CTH230 = [Th\text{-}230_s]^\circ \equiv [Th\text{-}230]^\circ$

$CTH232 = [Th\text{-}232_s]^\circ$

Figure 1.3.7.3
Calculate MF of
Th-230 to Th-232
(THINTH)

March 13, 1986

THINTH MF OF TH-230 TO TH-232 [-]. Mole fraction (more accurately, fraction) of Th-230 to Th-232 in the unsaturated zone.


1.3.7.3
MF of Th-230
to Th-232
(THINTH)
1.3.3
Init. Solid Phase
Conc. of Th-230
(CTH230)
1.3.7.1
Init. Solid Phase
Conc. of Th-232
(CTH232)

THINTH = CTH230 / CTH232 [-] (Eq. A.2.6)

Unit check: mol m^{-3} / (mol m^{-3}) = [-]

Note: Symbols used in Appendix A:

THINTH = $x_{Th/Th}$

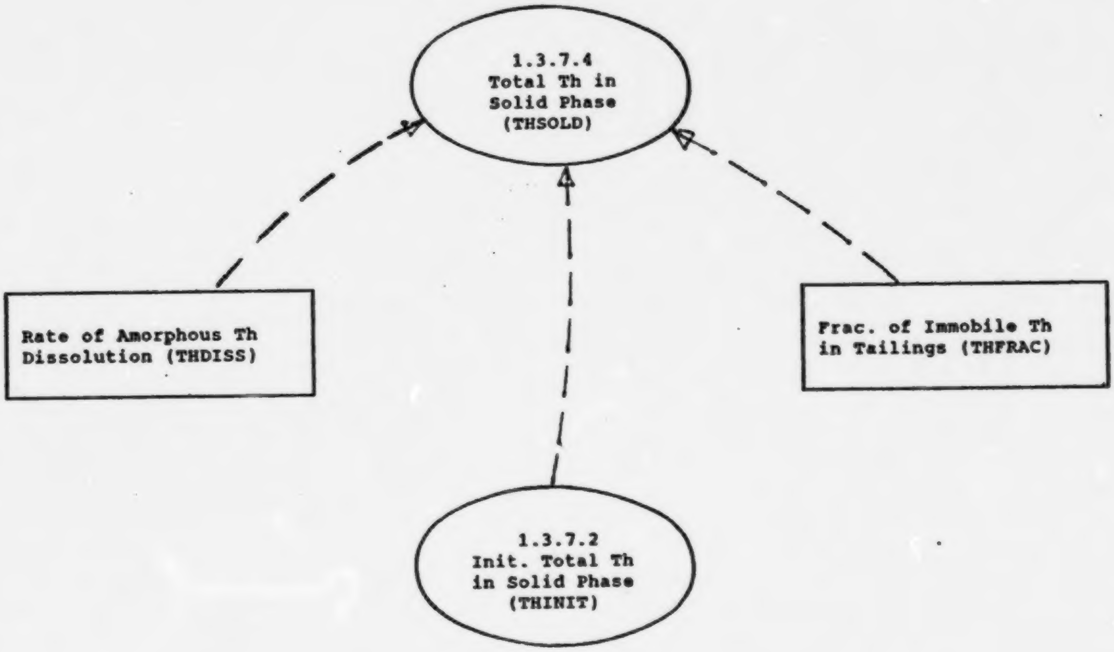
CTH230 = $[Th\text{-}230_s]^\circ \equiv [Th\text{-}230]^\circ$

CTH232 = $[Th\text{-}232_s]^\circ$

Figure 1.3.7.4
Calculate Total Th
in Solid Phase
(THSOLD)

March 13, 1986

THSOLD TOTAL TH IN SOLID PHASE [mol m^{-3}]. Concentration of total Th in the solid phase of the unsaturated zone.


1.3.7.4
Total Th in
Solid Phase
(THSOLD)
Rate of Amorphous Th
Dissolution (THDISS)
Frac. of Immobile Th
in Tailings (THFRAC)
1.3.7.2
Init. Total Th
in Solid Phase
(THINIT)

THSOLD = THFRAC * THINIT + (1-THFRAC) * THINIT * EXP(-THDISS * TIME) [mol m^{-3}] (Eq. A.5.1)

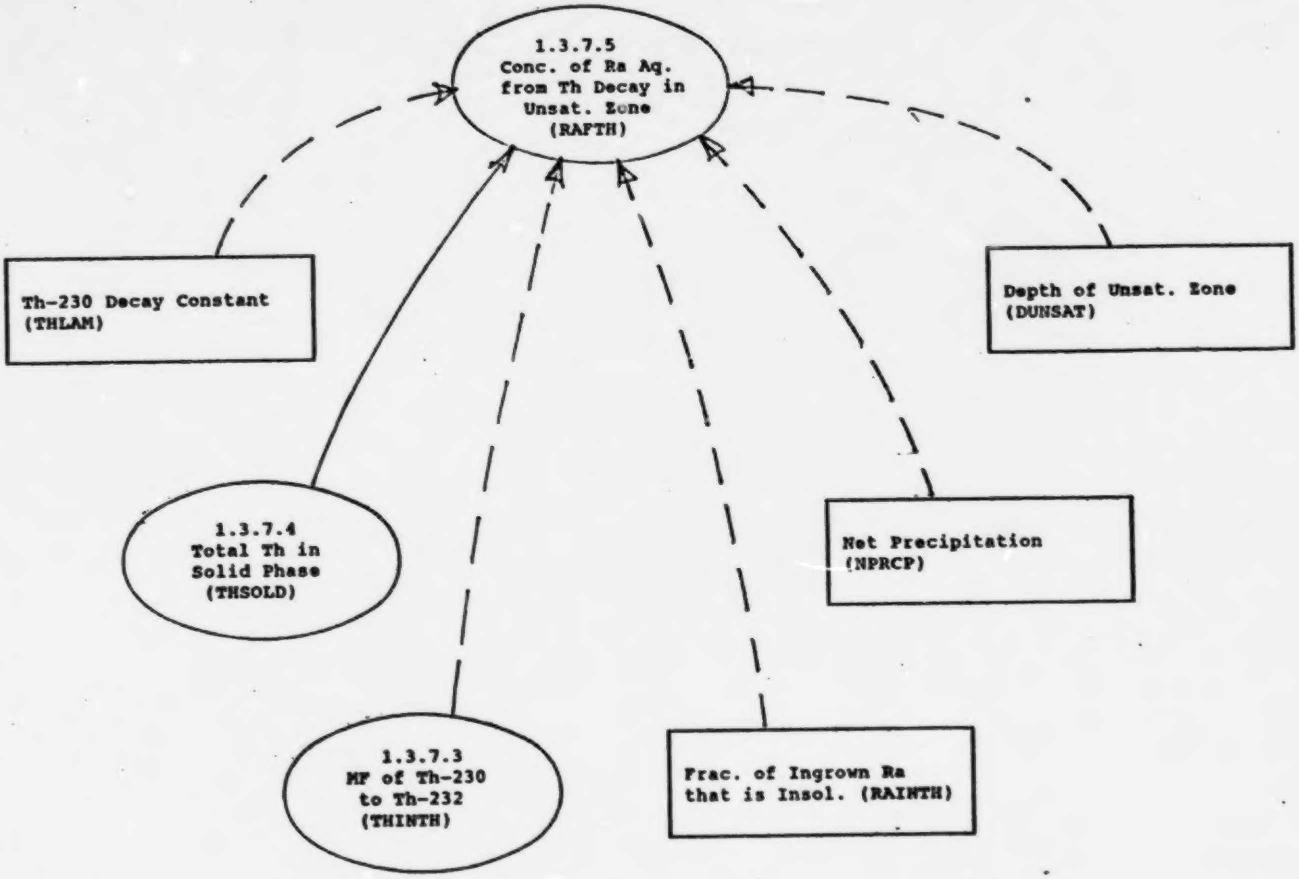
Unit check: (-) * mol m^{-3} + (-) * mol m^{-3} * EXP (a^{-1} * a) = [mol m^{-3}]

Note: Symbols used in Appendix A:

THSOLD = $[Th_s]$
THFRAC = X_{Th}
THINIT = $[Th]^o$
THDISS = k_{Th}
TIME = t

Figure 1.3.7.5
Calculate Conc. of
Ra Aq. from Th decay
in Unsat. Zone
(RAFTH)
March 13, 1986

RAFTH CONC. OF RA AQ FROM TH DECAY IN UNSAT. ZONE [mol m^{-3}]. Concentration of Ra-226 in the aqueous phase of the unsaturated zone, due to the decay of Th-230.


1.3.7.5
Conc. of Ra Aq.
from Th Decay in
Unsat. Zone
(RAFTH)
Th-230 Decay Constant
(THLAM)
Depth of Unsat. Zone
(DUNSAT)
1.3.7.4
Total Th in
Solid Phase
(THSOLD)
Net Precipitation
(NPRCP)
1.3.7.3
MF of Th-230
to Th-232
(THINTH)
Frac. of Ingrown Ra
that is Insol. (RAINTH)

$$\mathrm{RAFTH} = (1\text{-}\mathrm{RAINTH}) * \mathrm{THLAM} * \mathrm{THINTH} * \mathrm{THSOLD} * \frac{\mathrm{DUNSAT}}{\mathrm{NPRCP}} \qquad [\mathrm{mol\ m^{-3}}] \qquad \text{(Eq. A.5.5)}$$

Unit check: $(-) * a^{-1} * (-) * \mathrm{mol\ m^{-3}} * m / (m\ a^{-1}) = [\mathrm{mol\ m^{-3}}]$

Notes: 1. Appendix A is not clear when it comes to using RAFTH. Page A-18 uses C_{Ra} and page A-23 and Table A.1 use $[Ra_L]_{\lambda w}$. They appear to be equivalent. See also the note for CRAAVG (Figure 1.5.1).

2. Symbols used in Appendix A:

THLAM = λ_{Th}
THSOLD = $[Th]_s$
THINTH = $x_{Th/Th}$
DUNSAT = λ_w

NPRCP = P
RAINTH = $x_{Ra/Th}$
RAFTH = $[Ra_L]_{\lambda w}$

Figure 1.4 (P. 1 of 2)
Calculate Radium
Reprecipitation
(FRACRA) (REPRA)

May 12, 1986

FRACRA	FRAC. OF RA IN REPRECIP. GYPSUM [-]. Mole fraction of Ra-226 associated with reprecipitated gypsum in the lower (gypsum-enriched) layer of the unsaturated zone, as a function of time.
REPRA	MASS OF RA IN REPREC. GYPSUM [mol m^{-2}]. Radium content of reprecipitated gypsum per unit area in the bottom (gypsum-enriched) layer of the unsaturated zone, as a function of time and depth. Intermediate parameter introduced in analysis to simplify data flow diagrams.

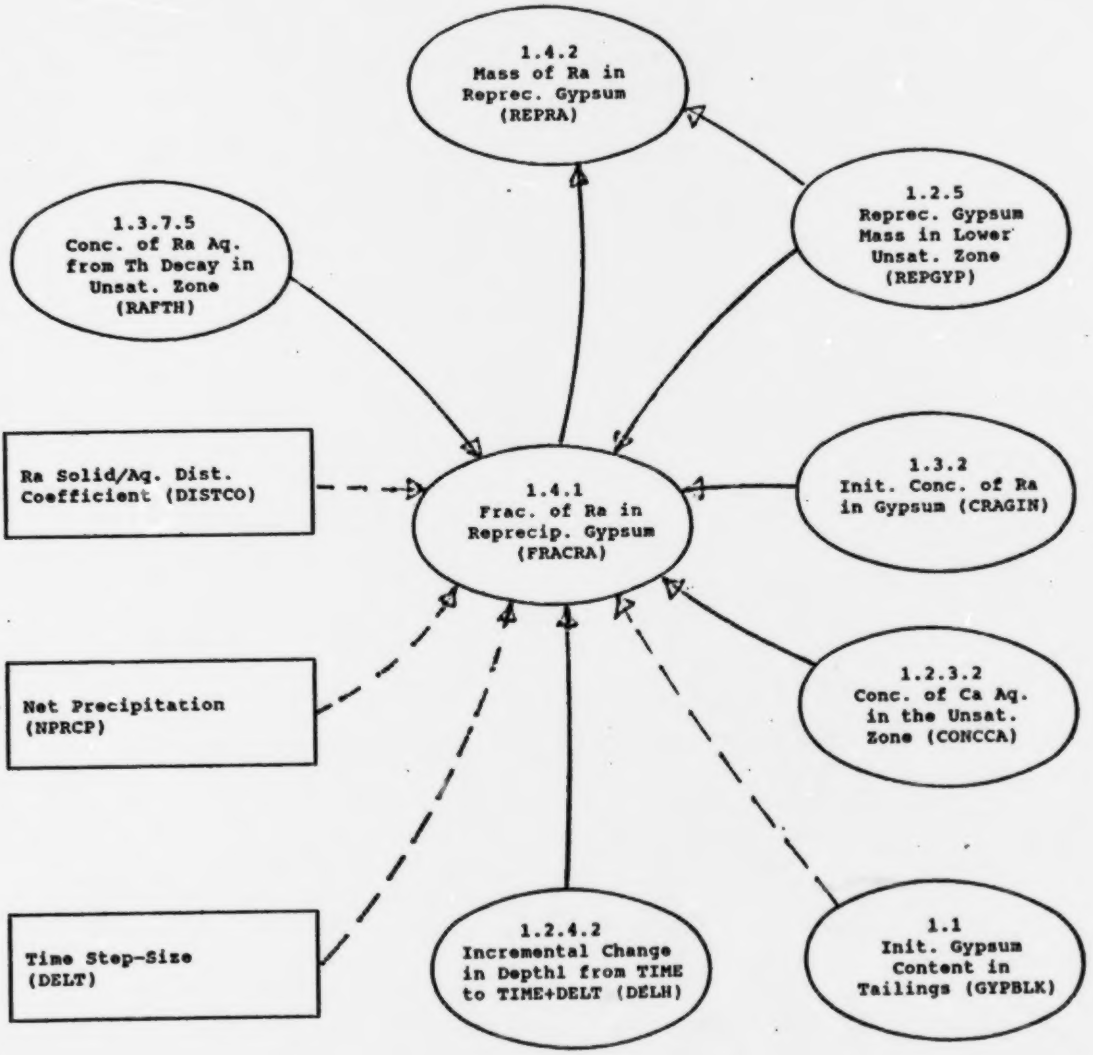

1.4.2
Mass of Ra in
Reprec. Gypsum
(REPRA)
1.3.7.5
Conc. of Ra Aq.
from Th Decay in
Unsat. Zone
(RAFTH)
1.2.5
Reprec. Gypsum
Mass in Lower
Unsat. Zone
(REPGYP)
Ra Solid/Aq. Dist.
Coefficient (DISTCO)
1.4.1
Frac. of Ra in
Reprecip. Gypsum
(FRACRA)
1.3.2
Init. Conc. of Ra
in Gypsum (CRAGIN)
Net Precipitation
(NPRCP)
1.2.3.2
Conc. of Ca Aq.
in the Unsat.
Zone (CONCCA)
Time Step-Size
(DELT)
1.2.4.2
Incremental Change
in Depth1 from TIME
to TIME+DELT (DELH)
1.1
Init. Gypsum
Content in
Tailings (GYPBLK)

Figure 1.4
(Concluded)

REPRA = FRACRA * REPGYP

where FRACRA is calculated by solving the following finite difference equation for FRACRA = x(t + Δt).

$$g_o \overline{X}^o_{Ra/gyp} \ \Delta h + P \ C_{Ra} \ \Delta t = m(t+\Delta t) \ x(t+\Delta t) - m(t) * x(t)$$

$$+ PK\Delta t \left\{ \frac{C_T(t+\Delta t) \ x(t+\Delta t) - C_T(t) \ x(t)}{\ln \left(\dfrac{C_T(t+\Delta t) \ x(t+\Delta t)}{C_T(t) \ x(t)} \right)} \right\} \qquad \text{(Eq. A.3.24)}$$

Initial Condition: x(0) = CRAGIN/GYPBLK [-] (See Note 2)

Unit check:

REPRA: $[\text{mol m}^{-2}] = [-] * [\text{mol m}^{-2}]$

FRACRA: $[\text{mol m}^{-3}] * (-) * [\text{m}] + [\text{m a}^{-1}] \ [\text{mol m}^{-3}] \ [\text{a}] = [\text{mol m}^{-2}] * [-] - [\text{mol m}^{-2}] * [-]$

$$+ [\text{m a}^{-1}] * [-] * [\text{a}] * \left\{ \frac{[\text{mol m}^{-3}] * [-] - [\text{mol m}^{-3}] * [-]}{\ln [-]} \right\}$$

Notes: 1. The symbols used here and in Appendix A are:

FRACRA = $x(t) \equiv x_t \equiv x$

DISTCO = K

REPGYP = $m(t, \lambda_w - h) \equiv m(t)$

CONCCA = $C_T(t) \equiv C_T(t, Z=\lambda_w)$

NPRCP = P

RAFTH = $C_{Ra} = [Ra_L]_{\lambda w} \equiv [Ra(aq)]$

CRAGIN = $g_o \overline{X}^o_{Ra/Gyp} \equiv [Ra_s]^o_{gyp}$

TIME = t

DELT = Δt

DELH = Δh

2. The initial condition is not specified in Appendix A of SENES Vol. 3, but SENES in a conversation on 11/3/86 stated that x(0) = 0. This initial condition was later revised, in a conversation with SENES on 6/5/86, to x(0) = CRAGIN/GYPBLK (which is called RAFRNI in Table A.1 of SENES Vol. 3). It was also noted, in a conversation with SENES on 10/4/86, that REPRA=0 at t=0, even though it is also assumed that the surface layer of unsaturated zone does not contain gypsum (for use in the dust module).

Figure 1.5
Calculate Ra Conc.
in Mixing Zone

March 13, 1986

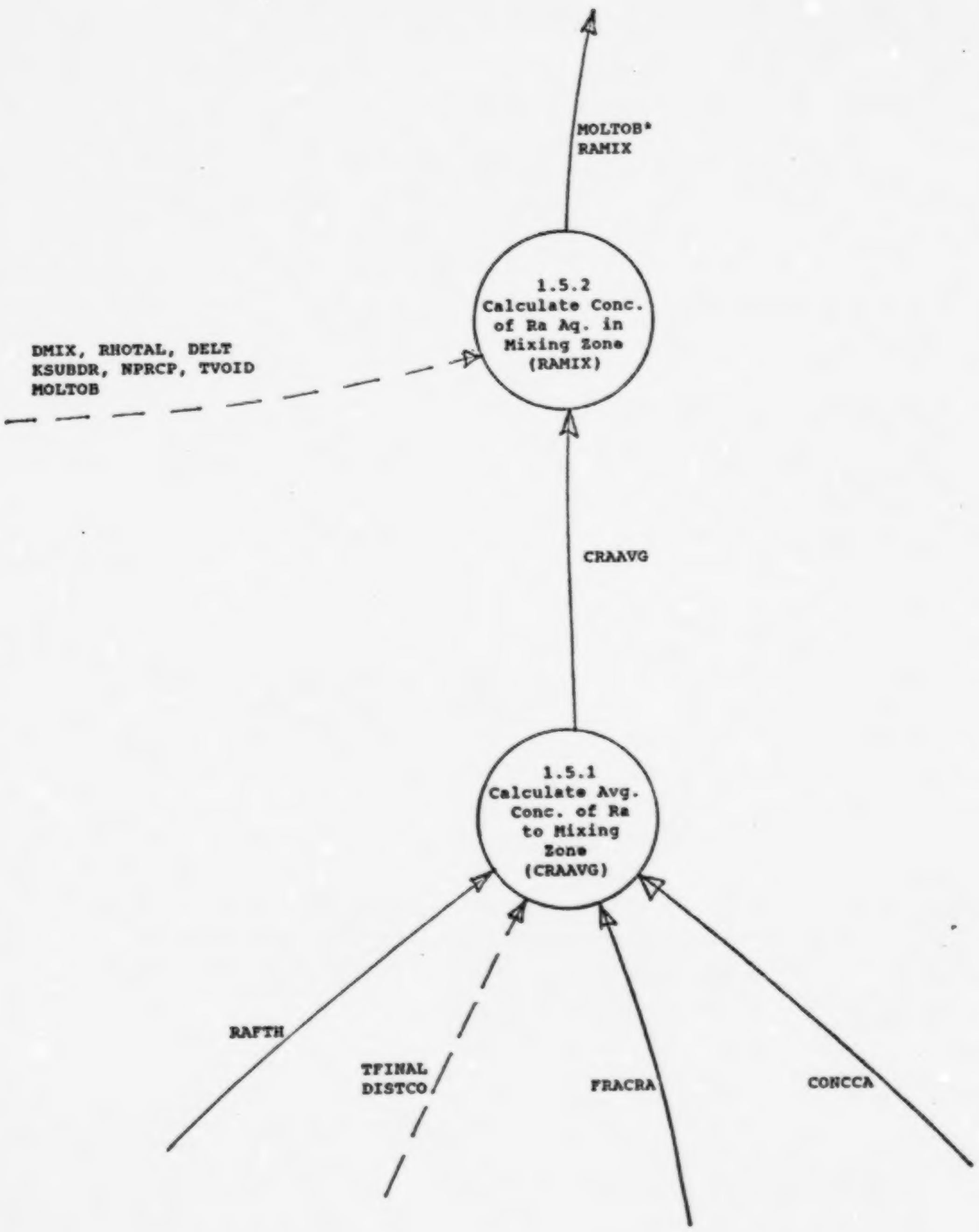
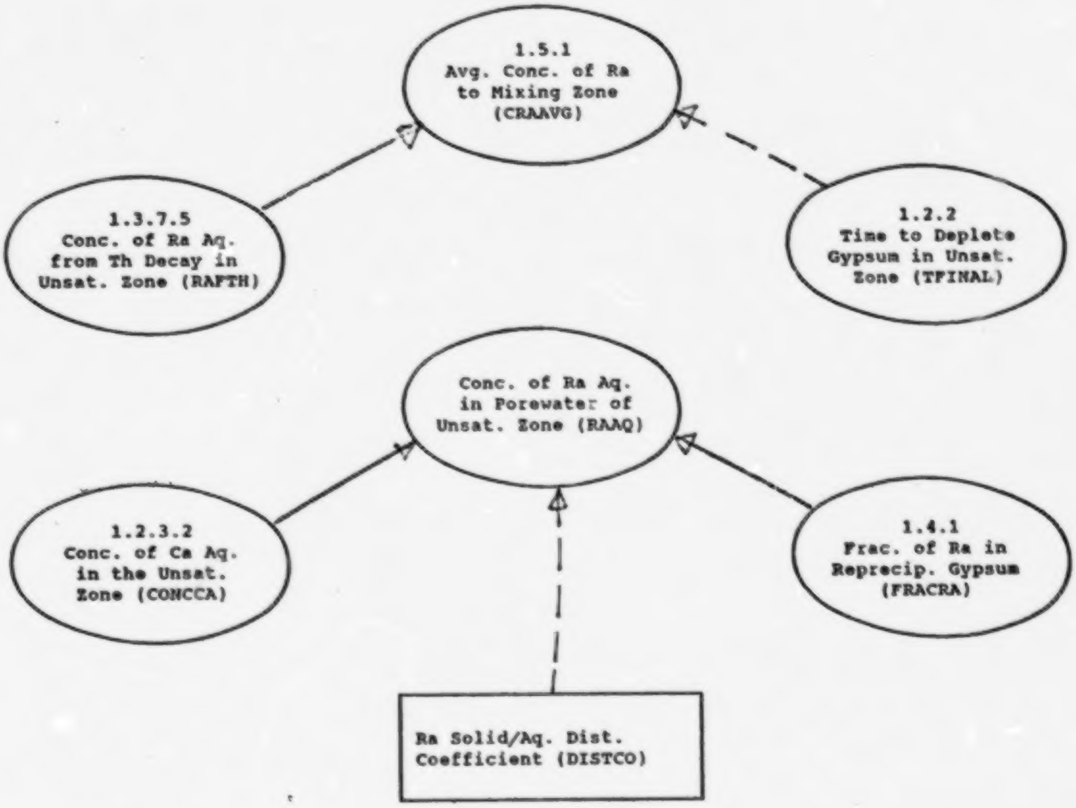

MOLTOB*
RAMIX
1.5.2
Calculate Conc.
of Ra Aq. in
Mixing Zone
(RAMIX)
DMIX, RHOTAL, DELT
KSUBDR, NPRCP, TVOID
MOLTOB
CRAAVG
1.5.1
Calculate Avg.
Conc. of Ra
to Mixing
Zone
(CRAAVG)
RAFTH
TFINAL
DISTCO
FRACRA
CONCCA

Figure 1.5.1
Calculate Avg. Conc. of Ra to Mixing Zone

May 12, 1986

CRAAVG AVG. CONC. OF RA TO MIXING ZONE [mol m^{-3}]. Average concentration of Ra-226 in the aqueous phase leaving bottom of the unsaturated zone and entering the mixing zone.


1.5.1
Avg. Conc. of Ra to Mixing Zone (CRAAVG)
1.3.7.5
Conc. of Ra Aq. from Th Decay in Unsat. Zone (RAFTH)
1.2.2
Time to Deplete Gypsum in Unsat. Zone (TFINAL)
Conc. of Ra Aq. in Porewater of Unsat. Zone (RAAQ)
1.2.3.2
Conc. of Ca Aq. in the Unsat. Zone (CONCCA)
1.4.1
Frac. of Ra in Reprecip. Gypsum (FRACRA)
Ra Solid/Aq. Dist. Coefficient (DISTCO)

CRAAVG = RAFTH (TIME $\geq$ TFINAL) [mol m^{-3}] (Eq. A.3.25)

CRAAVG = RAFTH + RAAQ (TIME < TFINAL) [mol m^{-3}] (See note 3)

where

RAAQ = DISTCO * FRACRA * CONCCA [mol m^{-3}] (Eq. A.3.2)

Unit check: [mol m^{-3}] = [mol m^{-3}]
[mol m^{-3}] = [-] * [-] * [mol m^{-3}]

Notes:
1. References to $[Ra]_{\lambda w}$, C_{Ra}, and $[Ra_L]_{\lambda w}$ are confusing. Only Eq. A.3.25 is available.
2. Symbols used in Appendix A:
   - CRAAVG = $[Ra(aq)] \equiv \overline{[Ra]}_{\lambda w}$ DISTCO = K
   - RAFTH = C_{Ra} = $[Ra_L]_{\lambda w}$ = $[Ra(t)_L]_{\lambda w}$ FRACRA = $x \equiv x_t \equiv x(t)$
   - RAAQ = $[Ra^{2+}(aq)] \equiv [Ra^{2+}]_{liquid}$
   - CONCCA = $[Ca^{2+}(Aq)] \equiv [Ca^{2+}] \equiv [gyp(aq)] \equiv C_T \equiv C_T(t,z=\lambda_w)$
3. As confirmed by SENES in a conversation dated 10/2/86.

Figure 1.5.2 (P. 1 of 2)
Calculate Conc. of
Ra Aq. in Mixing
Zone (RAMIX)

April 11, 1986

RAMIX CONC. OF RA AQ IN MIXING ZONE [mol m^{-3}]. Concentration of Ra-226 in the aqueous phase of the mixing zone.

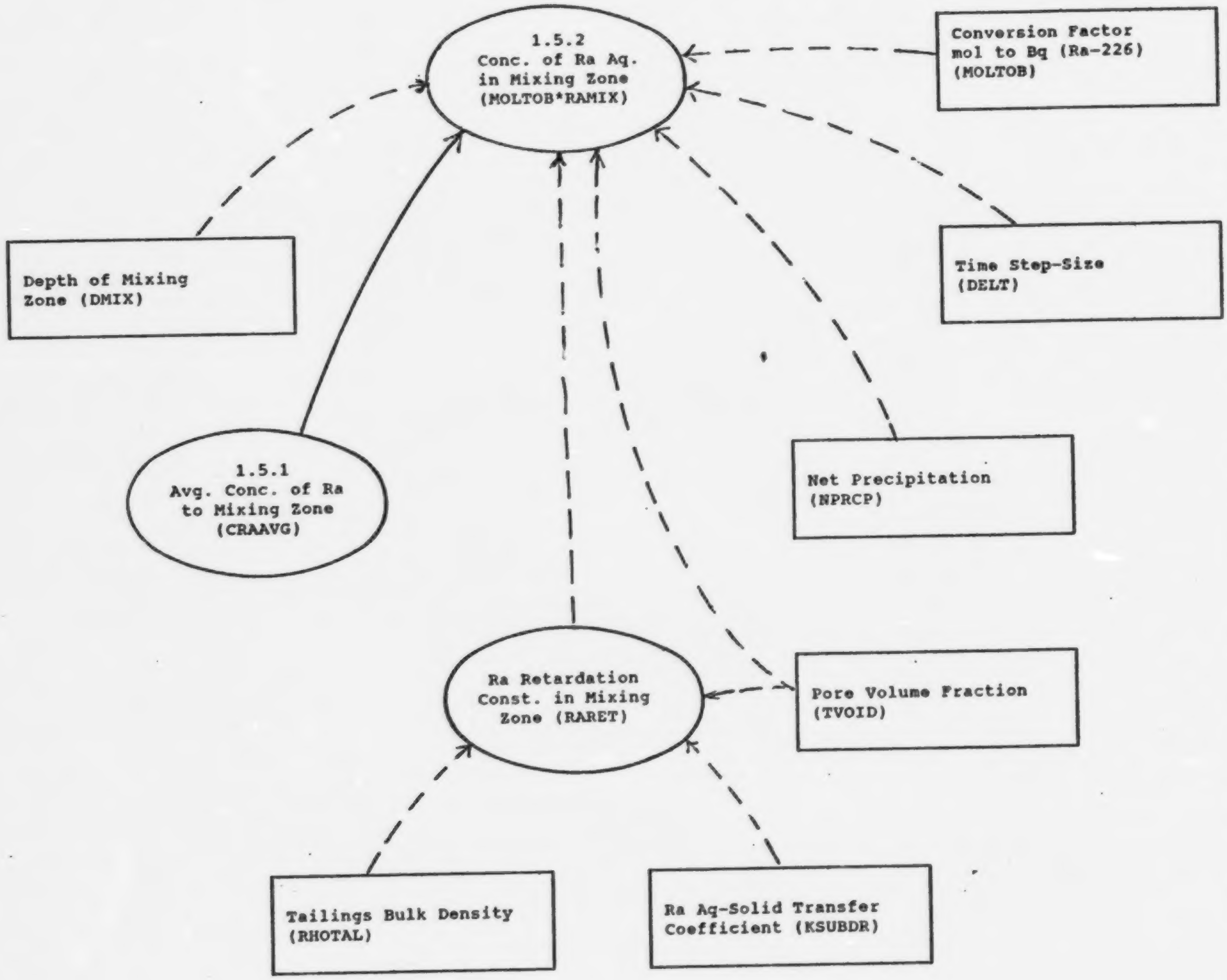

1.5.2
Conc. of Ra Aq.
in Mixing Zone
(MOLTOB*RAMIX)
Conversion Factor
mol to Bq (Ra-226)
(MOLTOB)
Depth of Mixing
Zone (DMIX)
Time Step-Size
(DELT)
1.5.1
Avg. Conc. of Ra
to Mixing Zone
(CRAAVG)
Net Precipitation
(NPRCP)
Ra Retardation
Const. in Mixing
Zone (RARET)
Pore Volume Fraction
(TVOID)
Tailings Bulk Density
(RHOTAL)
Ra Aq-Solid Transfer
Coefficient (KSUBDR)

Figure 1.5.2
Concluded

Solve the following f.d. equation for RAMIX (TIME+DELT)

RAMIX (TIME+DELT) = CRAAVG (TIME) + [RAMIX (TIME) − CRAAVG (TIME)]
* EXP [−NPRCP * DELT / (TVOID * DMIX * RARET)]
$[\mathrm{mol\ m^{-3}}]$ (Eq. A.7.11, see Note 3)

where

RARET = 1 + KSUBDR * RHOTAL / TVOID [−]

Unit check: RAMIX: mol $\mathrm{m^{-3}}$ + [mol $\mathrm{m^{-3}}$ − mol $\mathrm{m^{-3}}$] * EXP [m $\mathrm{a^{-1}}$ * a / ([−] * m)] = [mol $\mathrm{m^{-3}}$]
RARET: (−) + $\mathrm{m^3\ g^{-1}}$ * g $\mathrm{m^{-3}}$ / (−) = [−]

Equation check: for DELT=0, RAMIX (TIME) = RAMIX (TIME)

Notes: 1. Symbols used in Appendix A:

RAMIX = $[Ra]_t$ = $[Ra]_{mix}$
TIME = t
NPRCP = P
CRAAVG = $[\overline{Ra}]_{\lambda w}$ = [Ra(aq)]

DELT = Δt
KSUBDR = $K_{D,Ra}$
TVOID = ϵ
RHOTAL = ρ_T

2. It is assumed that RAMIX is also equivalent to the variable: "concentration of radium in the contaminated surface runoff from the tailings area", which is referred to in Appendix C on P. C-2.
3. Equation (A.7.11) is incorrect. A corrected version was supplied by SENES in a letter to B. Goodwin dated 12 February 1986.
4. RAMIX is required elsewhere in units of [Bq $\mathrm{m^{-3}}$]. It is converted using the following expression:

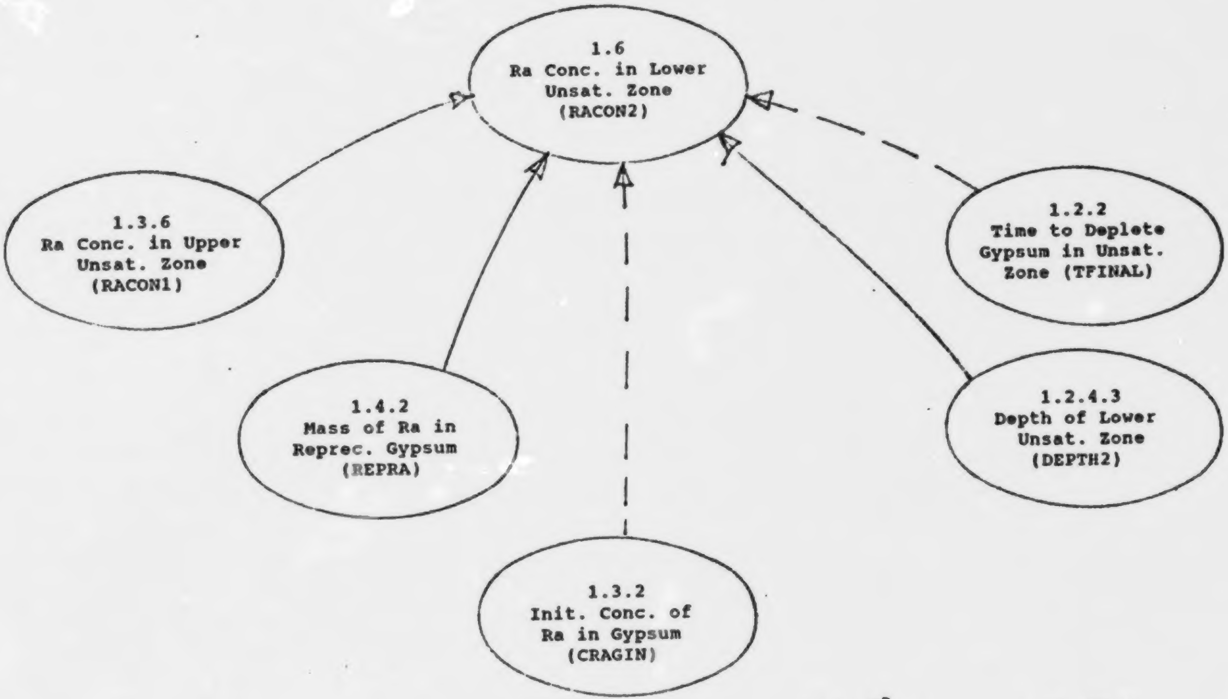
   RAMIX * MOLTOB (mol $\mathrm{m^{-3}}$) * (Bq $\mathrm{mol^{-1}}$) = [Bq $\mathrm{m^{-3}}$]

   where MOLTOB is a conversion factor for Ra-226.
5. The initial condition was originally assumed to be RAMIX(0)=0 (B. Goodwin). In a conversation with SENES on 10/4/86, it was noted that the assumption: RAMIX(0) = CRAAVG(0) is used.

Figure 1.6
Calculate Ra Conc.
in Lower Unsat. Zone
(RACON2)

March 13, 1986

RACON2 RA CONC. IN LOWER UNSAT. ZONE [mol m^{-3}]. Concentration of Ra-226 in the solid phase in the lower (gypsum-enriched) layer of the unsaturated zone.


1.6
Ra Conc. in Lower
Unsat. Zone
(RACON2)
1.3.6
Ra Conc. in Upper
Unsat. Zone
(RACON1)
1.2.2
Time to Deplete
Gypsum in Unsat.
Zone (TFINAL)
1.4.2
Mass of Ra in
Reprec. Gypsum
(REPRA)
1.2.4.3
Depth of Lower
Unsat. Zone
(DEPTH2)
1.3.2
Init. Conc. of
Ra in Gypsum
(CRAGIN)

$$\text{RACON2} = \text{RACON1} \qquad (\text{TIME} \geq \text{TFINAL}) \qquad [\text{mol m}^{-3}]$$
(No Eqn.)

$$\text{RACON2} = \text{RACON1} + \text{CRAGIN} + \frac{\text{REPRA}}{\text{DEPTH2}} \qquad (\text{TIME} < \text{TFINAL}) \qquad [\text{mol m}^{-3}]$$
(Eq. A.6.2, simplified for clarity)

Unit check: $\text{mol m}^{-3} + \text{mol m}^{-3} + \text{mol m}^{-2} / \text{m} = [\text{mol m}^{-3}]$

Notes:

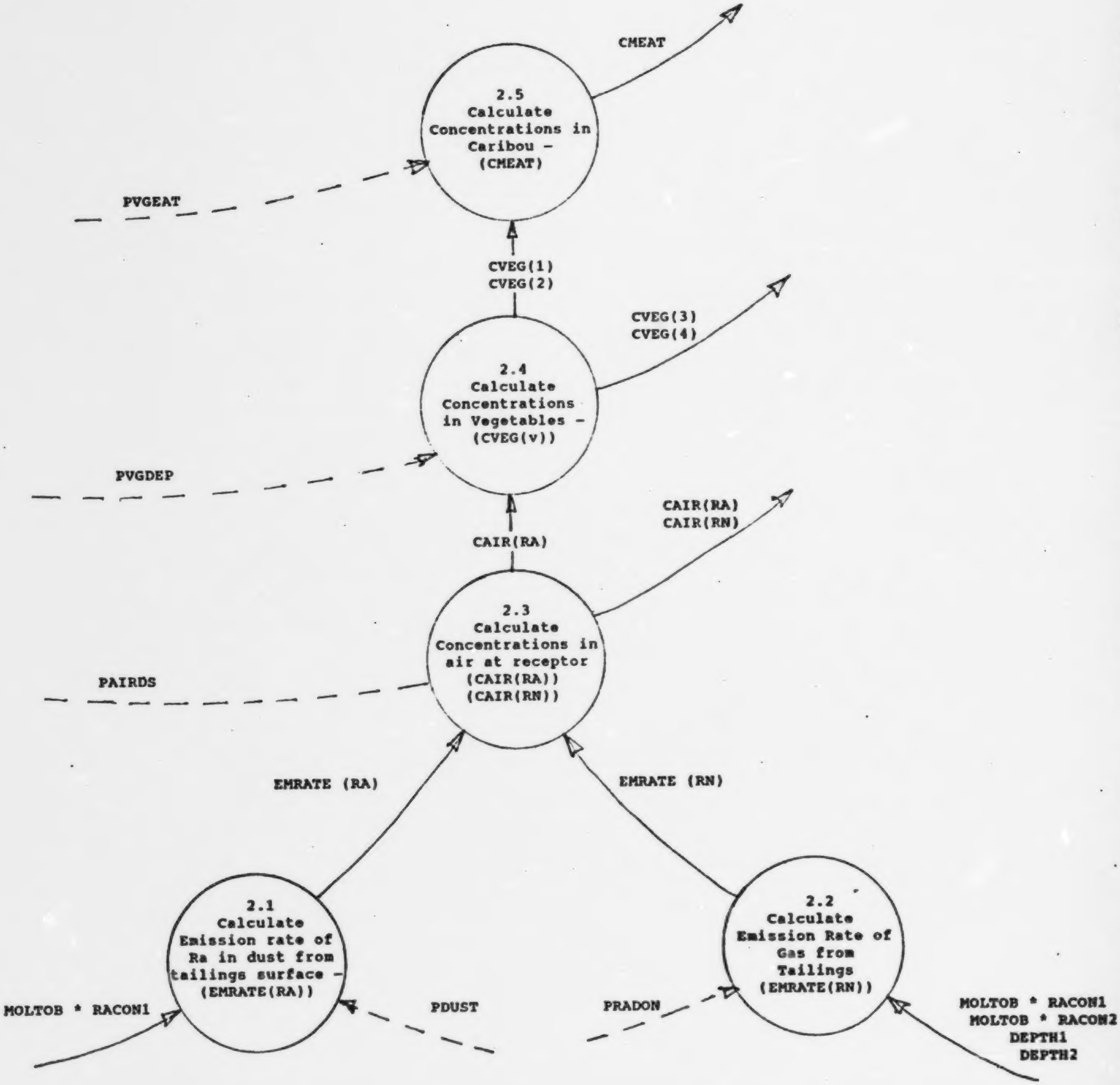
1. Considerable simplification of Equation (A.6.2) has taken place, using existing parameter definitions, for CRAGIN, REPRA and DEPTH2.
2. Symbols used in Appendix A:

RACON1 = $[Ra_s]_1$
RACON2 = $[Ra_s]_2$
CRAGIN = $g_o * \overline{X}^{o}_{Ra/gyp} \equiv [Ra_s]^{o}_{gyp}$
TFINAL = t_f
REPRA = $x(t) * m(t, \lambda_w - h)$

FRACRA = $x \equiv x(t) \equiv x_t$
DEPTH2 = $\lambda_w - h$
DEPTH1 = h
DUNSAT = λ_w

Figure 2.0
Calculate Concentrations from Atmospheric Pathways

February 25, 1986


CMEAT
2.5
Calculate
Concentrations in
Caribou -
(CMEAT)
PVGEAT
CVEG(1)
CVEG(2)
CVEG(3)
CVEG(4)
2.4
Calculate
Concentrations
in Vegetables -
(CVEG(v))
PVGDEP
CAIR(RA)
CAIR(RN)
CAIR(RA)
2.3
Calculate
Concentrations in
air at receptor
(CAIR(RA))
(CAIR(RN))
PAIRDS
EMRATE (RA)
EMRATE (RN)
2.1
Calculate
Emission rate of
Ra in dust from
tailings surface -
(EMRATE(RA))
2.2
Calculate
Emission Rate of
Gas from
Tailings
(EMRATE(RN))
MOLTOB * RACON1
PDUST
PRADON
MOLTOB * RACON1
MOLTOB * RACON2
DEPTH1
DEPTH2

Notes: PDUST, PRADON, PAIRDS, PVGDEP AND PVGEAT are clusters of input (sampled) parameters required by the bubbles they point to. Their contents are:

PAIRDS = XDIST, HSRC, VDISP, FWIND2, UWIND.

PDUST = SILT, FPREC, FWIND1, AEXPT, RHOTAL.

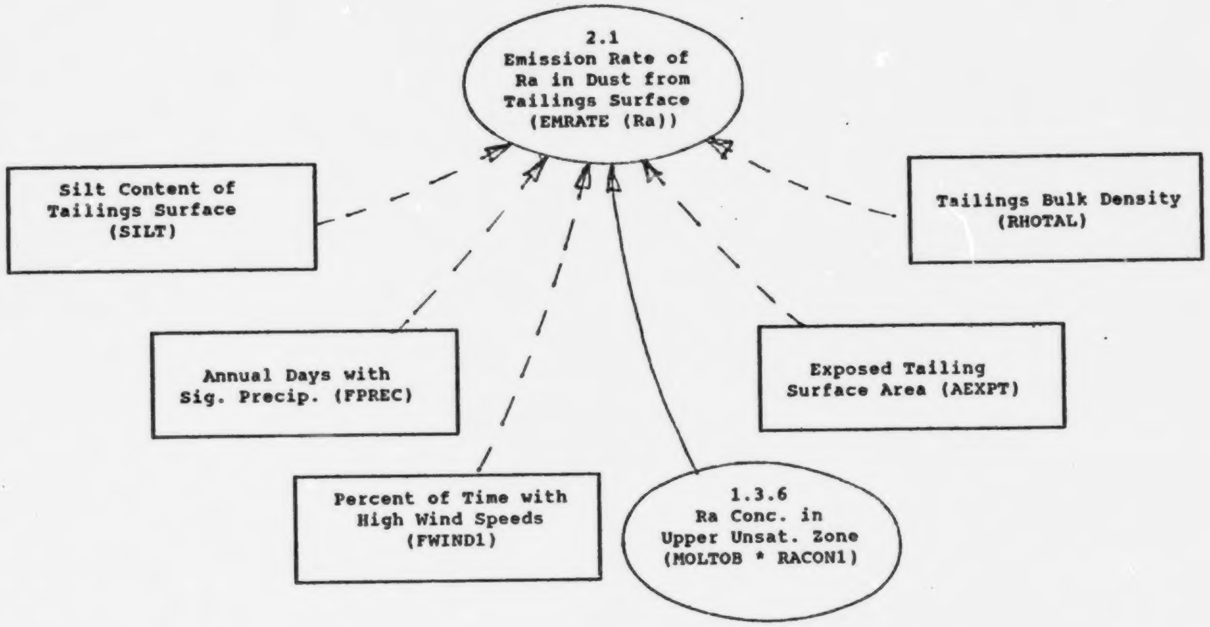
PRADON = ATAIL, RNLAM, EMANAT, RHOTAL, TVOID, MPCH2O.

PVGDEP = DEPVEL, FDEPSF(v), FDEPED(v), VEGLAM(v), DUREXP(v), YDEN(v); v=1,2,3,4.

PVGEAT = FIR, FBI, FAUT(v); v = 1,2.

Figure 2.1
Calculate Emission Rate
of Ra in Dust from
Tailings Surface
(EMRATE(RA))
February 25, 1986

EMRATE(RA) EMISSION RATE OF RA IN DUST FROM TAILINGS SURFACE [Bq s^{-1}]. Emission rate of Ra-226 in dust from the tailings surface, as a function of time.


2.1
Emission Rate of
Ra in Dust from
Tailings Surface
(EMRATE (Ra))
Silt Content of
Tailings Surface
(SILT)
Tailings Bulk Density
(RHOTAL)
Annual Days with
Sig. Precip. (FPREC)
Exposed Tailing
Surface Area (AEXPT)
Percent of Time with
High Wind Speeds
(FWIND1)
1.3.6
Ra Conc. in
Upper Unsat. Zone
(MOLTOB * RACON1)

$$EMRATE\ (RA) = \frac{2.2 \times 10^{-6} * SILT * (365-FPREC) * FWIND1 * (MOLTOB * RACON1) * AEXPT}{1.5 * 235 * 15 * RHOTAL}$$

[Bq s^{-1}] (Eq. B.1.1; see Note 1))

Unit check: Assume constants have units shown in Note 2.
$[g\ m^{-2}s^{-1} * \% * d * \% * Bq\ mol^{-1} * mol\ m^{-3} \times m^{2}] / [\% * d * \% * gm^{-3}] = [Bq\ s^{-1}]$

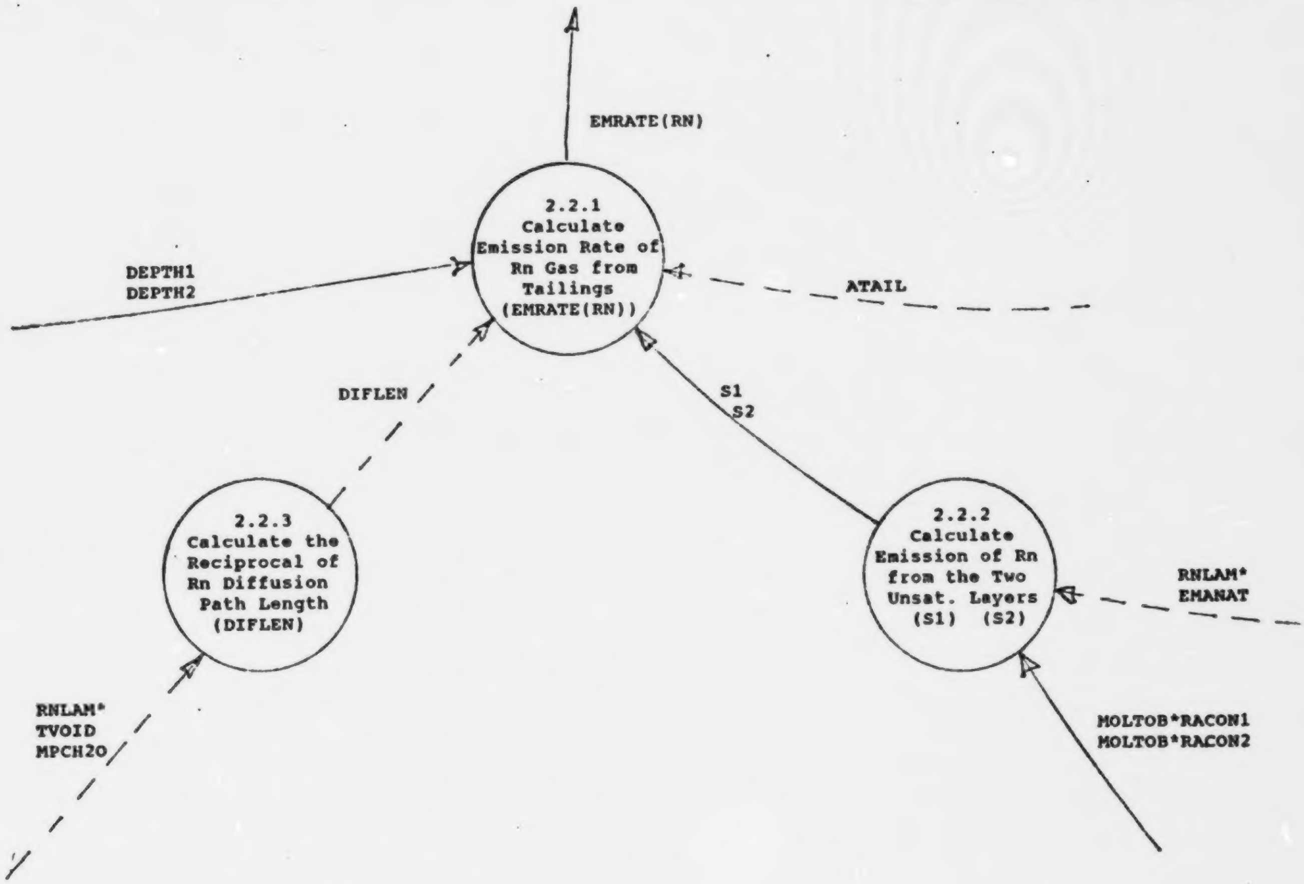
Notes: 1. Unit inconsistency. Description of Eq. (B.1.1) shows RACON1 in [Bq m^{-3}], Table B.1 in [Bq g^{-1}]; but Eq. (A.6.1) in [mol. m^{-3}]. Resolved in discussions with SENES on 10/2/86. RACON1 must be multiplied by MOLTOB to convert RACON1 from [mol m^{-3}] to [Bq m^{-3}].

2. The constants used in the above equation are described in SENES Vol. 3, and have the following units

2.2×10^{-6} [g $m^{-2}s^{-1}$] (p B-2)
1.5 [%] (assumed)
365, 235 [d] (assumed)
15 [%] (assumed)

Figure 2.2
Calculate Emission Rate
of Rn Gas from Tailings

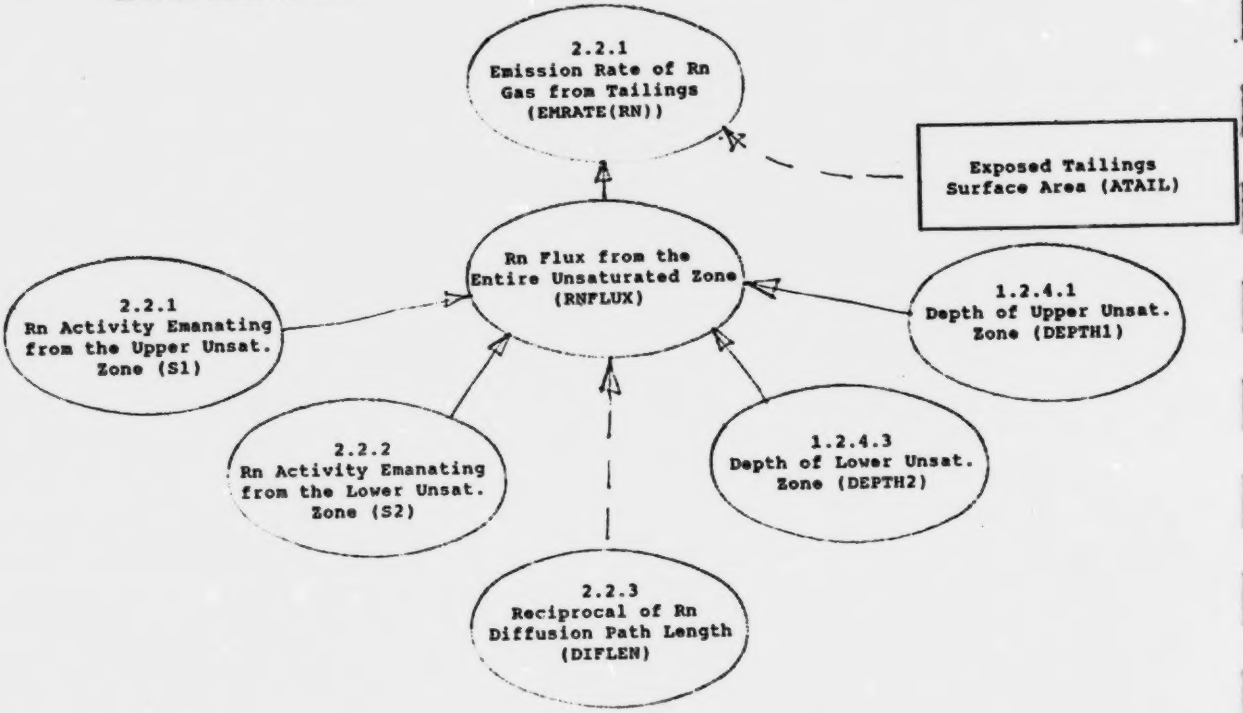
February 24, 1986


EMRATE(RN)
2.2.1
Calculate
Emission Rate of
Rn Gas from
Tailings
(EMRATE(RN))
DEPTH1
DEPTH2
ATAIL
DIFLEN
S1
S2
2.2.3
Calculate the
Reciprocal of
Rn Diffusion
Path Length
(DIFLEN)
2.2.2
Calculate
Emission of Rn
from the Two
Unsat. Layers
(S1) (S2)
RNLAM*
EMANAT
RNLAM*
TVOID
MPCH2O
MOLTOB*RACON1
MOLTOB*RACON2

Notes:
1. RHOTAL is not required, for the reasons noted in figure 2.2.2.
2. MOLTOB is required to correct the units of RACON1 and RACON2, as noted in Figure 2.2.2.
3. TVOID is not really required, as noted in Figure 2.2.3, but is included to maintain consistency with SENES Vol. 3.

Figure 2.2.1
Calculate
Emission Rate of
Rn Gas from Tailings
(EMRATE(RN))
February 25, 1986

EMRATE(RN) EMISSION RATE OF RN GAS FROM TAILINGS [Bq s^{-1}].
Emission rate of Rn-222 gas from the tailings unsaturated zone (upper and lower layers) as a function of time.


2.2.1
Emission Rate of Rn
Gas from Tailings
(EMRATE(RN))
Exposed Tailings
Surface Area (ATAIL)
Rn Flux from the
Entire Unsaturated Zone
(RNFLUX)
2.2.1
Rn Activity Emanating
from the Upper Unsat.
Zone (S1)
1.2.4.1
Depth of Upper Unsat.
Zone (DEPTH1)
2.2.2
Rn Activity Emanating
from the Lower Unsat.
Zone (S2)
1.2.4.3
Depth of Lower Unsat.
Zone (DEPTH2)
2.2.3
Reciprocal of Rn
Diffusion Path Length
(DIFLEN)

$$\mathrm{EMRATE(RN) = RNFLUX * ATAIL}$$ [Bq s^{-1}] (Eq. on Page B-7)

$$\mathrm{RNFLUX} = \frac{-\mathrm{S1}}{\mathrm{DIFLEN}} + \frac{1}{\mathrm{DIFLEN * COSH\,(DIFLEN*DEPTH1)}} * \frac{\mathrm{[1 + COTH(DIFLEN*DEPTH2)] * S1 * EXP(DIFLEN * DEPTH1) - S1 + S2}}{\mathrm{TANH(DIFLEN*DEPTH1) + COTH(DIFLEN*DEPTH2)}}$$

[Bq $m^{-2}s^{-1}$] (Eq. B.2.4; see Note

Unit check: EMRATE: Bq m^{-2} s^{-1} · m^2 = [Bq s^{-1}]

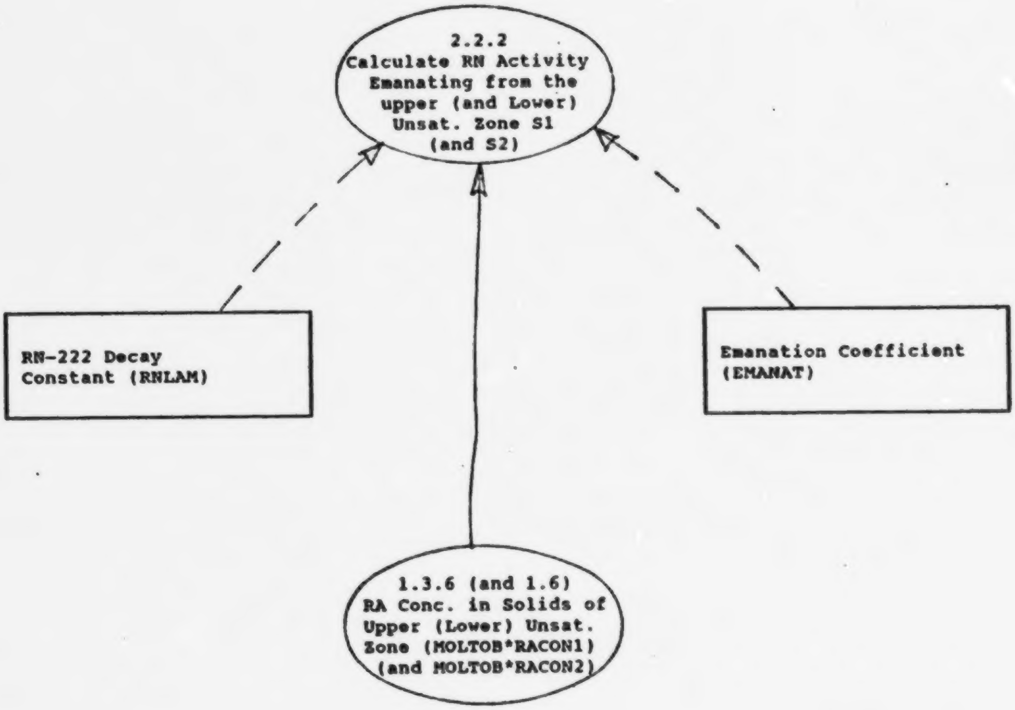
DIFLEN * DEPTHX: m^{-1} · m = [-]

RNFLUX: Bq m^{-3} s^{-1} * m +... = [Bq m^{-2} s^{-1}]

Note: The factor D_i in Eq. B.2.4 has been cancelled with D_i from S1 and S2.

Figure 2.2.2
Calculate Emission of
Rn from the Two
Unsat. Zones (S1)
(S2)
February 25, 1986

S1 (and S2)RN ACTIVITY EMANATING FROM THE UPPER (LOWER) UNSAT. ZONE [Bq m^{-3} s^{-1}]. Proportional to the rate of emission of Rn-222 per unit area from the upper gypsum-depleted (and lower gypsum-enriched) layer of the unsaturated zone.


2.2.2
Calculate RN Activity Emanating from the upper (and Lower) Unsat. Zone S1 (and S2)
RN-222 Decay Constant (RNLAM)
Emanation Coefficient (EMANAT)
1.3.6 (and 1.6) RA Conc. in Solids of Upper (Lower) Unsat. Zone (MOLTOB*RACON1) (and MOLTOB*RACON2)

$$S1 = MOLTOB * RACON1 * RNLAM * EMANAT \qquad [Bq\ m^{-3}\ s^{-1}]$$
$$S2 = MOLTOB * RACON2 * RNLAM * EMANAT$$

(equation on Page B-6.; see Notes 1 and 2)

Unit check: $Bq\ mol^{-1} * mol\ m^{-3} \times s^{-1} \times [-] = [Bq\ m^{-3} s^{-1}]$

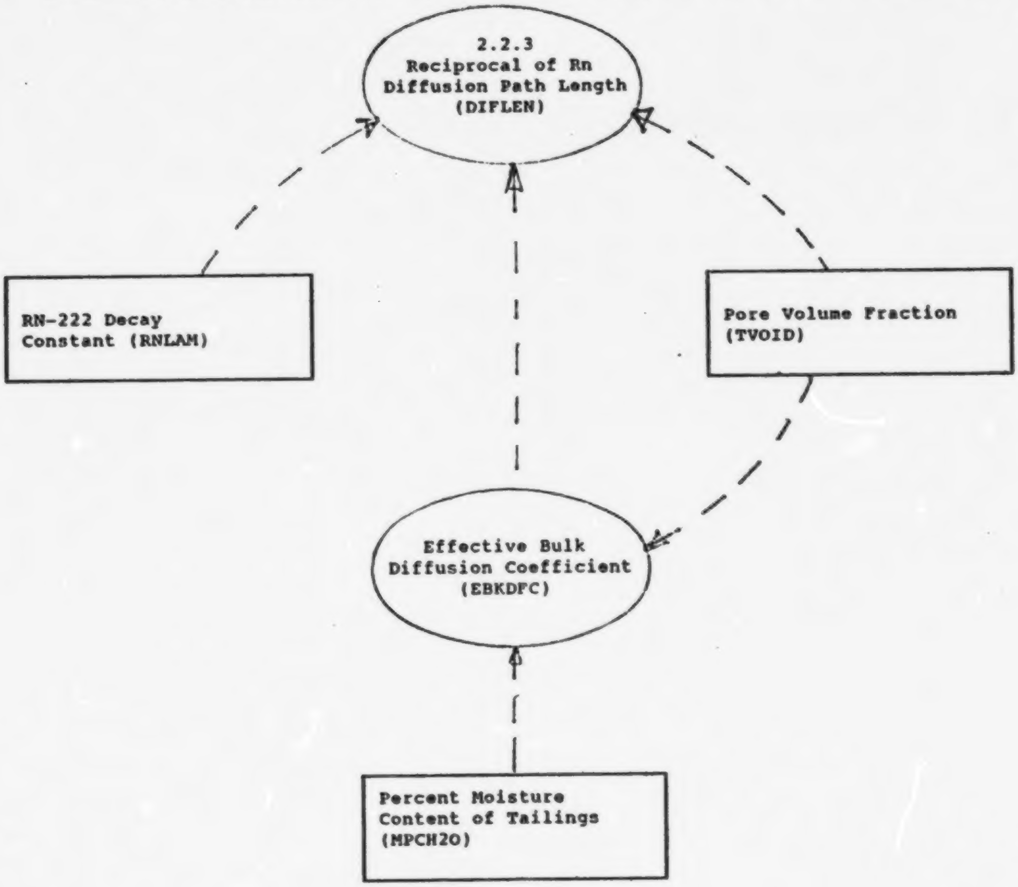
Notes 1. Unit inconsistency using units for RACON1 and RACON2 given in SENES Vol. 3 in equations (A.6.1), (A.6.2), near (B.2.1) and in Table B.2. Following discussions with SENES on 10/2/86, it was noted that Ra concentrations must be in [Bq m^{-3}]. This means that:

- RACON1 and RACON2 must be multiplied by the conversion factor MOLTOB to convert RACON1 and RACON2 from [mol m^{-3}] to [Bq m^{-3}].
- RHOTAL is not used in the above equation.

2. The above equations have been further modified from those used in SENES Vol. 3. The factor D_i has been dropped out, since it cancels with the factor D_i in Eq. B.2.4, the only place S1 and S2 are used.

Figure 2.2.3
Calculate the Reciprocal
of Rn Diffusion
Path Length
(DIFLEN)
February 25, 1986

DIFLEN RECIPROCAL OF RN DIFFUSION PATH LENGTH [m^{-1}]. Reciprocal of the Rn-222 diffusion path length.


2.2.3
Reciprocal of Rn
Diffusion Path Length
(DIFLEN)
RN-222 Decay
Constant (RNLAM)
Pore Volume Fraction
(TVOID)
Effective Bulk
Diffusion Coefficient
(EBKDFC)
Percent Moisture
Content of Tailings
(MPCH2O)

EBKDFC = 0.106 x 10^{-4} * TVOID * EXP (-0.261 * MPCH2O) [$m^2 s^{-1}$] (Eq. near B.2.1)

DIFLEN = (RNLAM * TVOID / EBKDFC)$^{\frac{1}{2}}$ [m^{-1}] (Eq. near B.2.2)

Unit Check: not possible for EBKDFC, since the units of the constants are not given

DIFLEN: $[s^{-1} \cdot m^3 m^{-3} / m^2 s^{-1}]^{\frac{1}{2}} = m^{-1}$

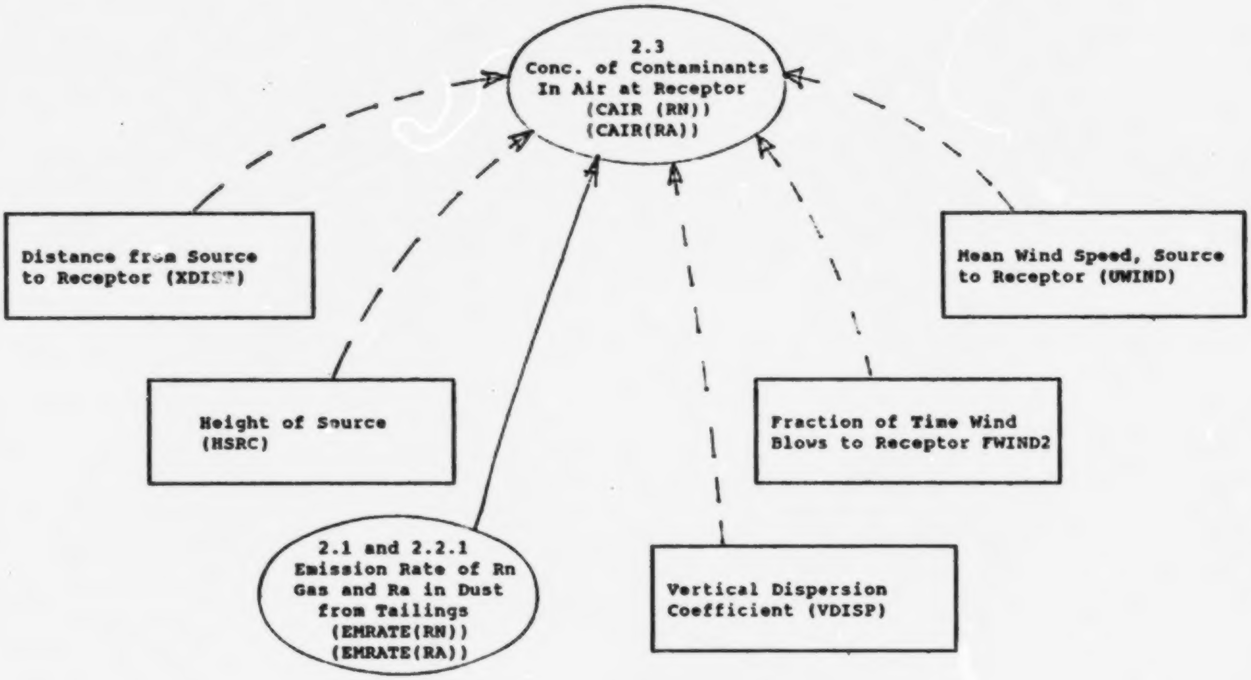
Note: 1. Combining the above,

DIFLEN = [RNLAM / [0.106 x 10^{-4} * EXP (-0.261 * MPCH2O)]]$^{\frac{1}{2}}$
which does not depend on TVOID. This latter equation should not be used, however, so as to maintain consistency with SENES Vol. 3.

2. The constants used in EBKDFC have unknown dimensions.

Figure 2.3
Conc. of Contaminant
in Air at Receptor
(CAIR(RN))
(CAIR(RA)
February 25, 1986

CAIR(A)	CONC. OF CONTAMINANT (RA) IN AIR AT RECEPTOR [Bq m^{-3}]. Average annual concentration of Ra-226 associated with dust in air at a receptor (locally grown vegetables and/or a 'critical' man).
CAIR(RN)	CONC. OF CONTAMINANT (RN) IN AIR AT RECEPTOR [Bq m^{-3}]. Average annual concentration of Rn-222 gas in air at a receptor (a 'critical' man).


2.3
Conc. of Contaminants
In Air at Receptor
(CAIR (RN))
(CAIR(RA))
Distance from Source to Receptor (XDIST)
Mean Wind Speed, Source to Receptor (UWIND)
Height of Source (HSRC)
Fraction of Time Wind Blows to Receptor FWIND2
2.1 and 2.2.1
Emission Rate of Rn Gas and Ra in Dust from Tailings
(EMRATE(RN))
(EMRATE(RA))
Vertical Dispersion Coefficient (VDISP)

$$CAIR\ (I) = \frac{2.032 * FWIND2 * EMRATE(I)}{UWIND * VDISP * XDIST} * EXP \left\{ - \frac{(HSRC)^2}{2*(VDISP)^2} \right\} \quad [Bq\ m^{-3}]\ \text{(Eq. on Page D.1)}$$

where I = Ra or Rn.

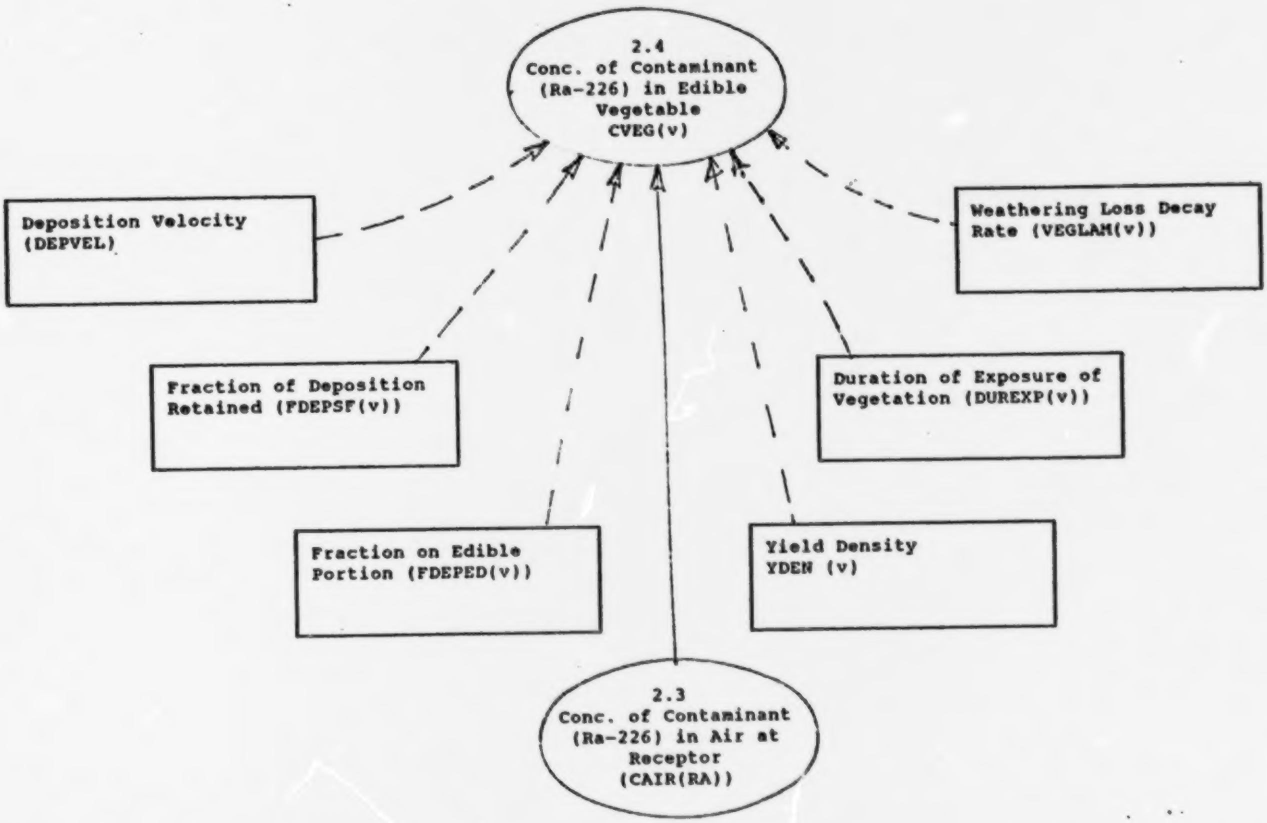
Unit check: $[-] * [-] * Bq\ s^{-1} / [m\ s^{-1} * m * m]\ EXP\ (m^2 / m^2) = [Bq\ m^{-3}]$

Note: The constant 2.032 has unknown units, but is assumed to be dimensionless in the unit check.

Figure 2.4
Calculate Concentrations
in Vegetables
(CVEG(v))

February 25, 1986

CVEG(v) CONC. OF CONTAMINANT (RA-226) IN EDIBLE VEGETABLE [Bq g^{-1} (wet weight)]. Concentration of Ra-226 in edible vegetables due to foliar retention of Ra-226 deposited from dust in the air. Four types of vegetation are considered (v=1,2,3,4=lichen, grass, above-ground and below-ground classes).


2.4
Conc. of Contaminant
(Ra-226) in Edible
Vegetable
CVEG(v)
Deposition Velocity
(DEPVEL)
Weathering Loss Decay
Rate (VEGLAM(v))
Fraction of Deposition
Retained (FDEPSF(v))
Duration of Exposure of
Vegetation (DUREXP(v))
Fraction on Edible
Portion (FDEPED(v))
Yield Density
YDEN (v)
2.3
Conc. of Contaminant
(Ra-226) in Air at
Receptor
(CAIR(RA))

$$\mathrm{CVEG(v) = CAIR(RA) * DEPVEL * FDEPSF(v) * FDEPED(v)} * \left\{ \frac{\mathrm{1 - EXP\ [-VEGLAM(v) * DUREXP(v)]}}{\mathrm{YDEN(v) * VEGLAM(v)}} \right\} \quad [\mathrm{Bq\ g^{-1}}] \quad \text{(Eq. E.1.1)}$$

Unit check: CVEG: $[\mathrm{Bq\ m^{-3} * m\ s^{-1} * (-) * (-)}] / [\mathrm{g\ m^{-2} * s^{-1}}] = [\mathrm{Bq\ g^{-1}}]$
EXP: $\mathrm{s^{-1} * s} = [-]$

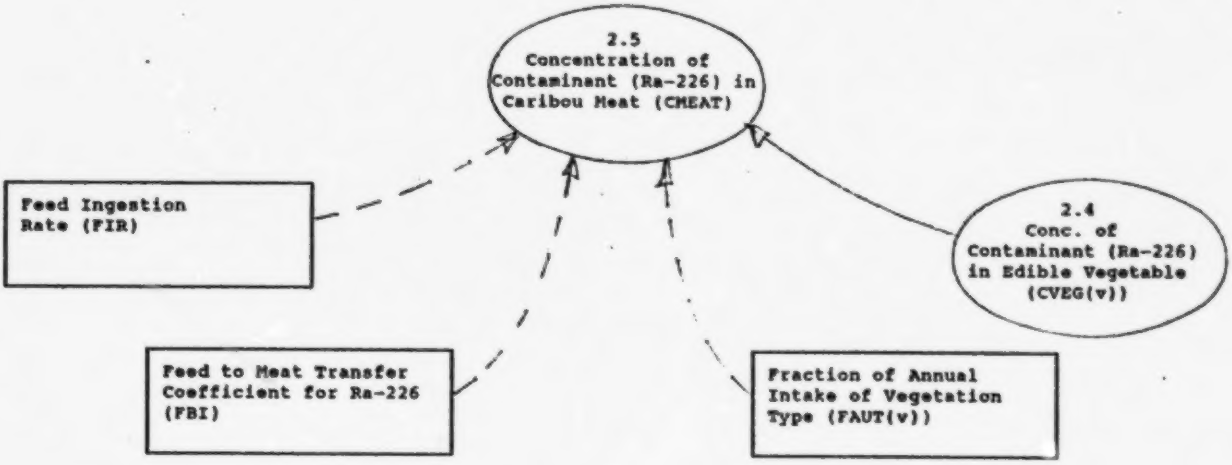
Note: 1. There are four types of vegetation, denoted by the index v. For v = 1,2,3,4, the types are lichen, grass, above-ground vegetables and below ground vegetables, respectively.

2. Vegetable types 1 and 2 are used to calculate concentrations of Ra in caribou meat. Vegetable types 3 and 4 are used to calculate dose.

Figure 2.5
Calculate Concentrations in Caribou (CMEAT)

February 28, 1986

CMEAT CONCENTRATION OF CONTAMINANT (RA-226) IN CARIBOU MEAT [Bq g^{-1} (wet weight)]. Average annual concentration of Ra-226 in caribou meat, due to the ingestion of lichen and grass (vegetation types 1 and 2) which have been contaminated by air deposition.


2.5
Concentration of Contaminant (Ra-226) in Caribou Meat (CMEAT)
Feed Ingestion Rate (FIR)
2.4
Conc. of Contaminant (Ra-226) in Edible Vegetable (CVEG(v))
Feed to Meat Transfer Coefficient for Ra-226 (FBI)
Fraction of Annual Intake of Vegetation Type (FAUT(v))

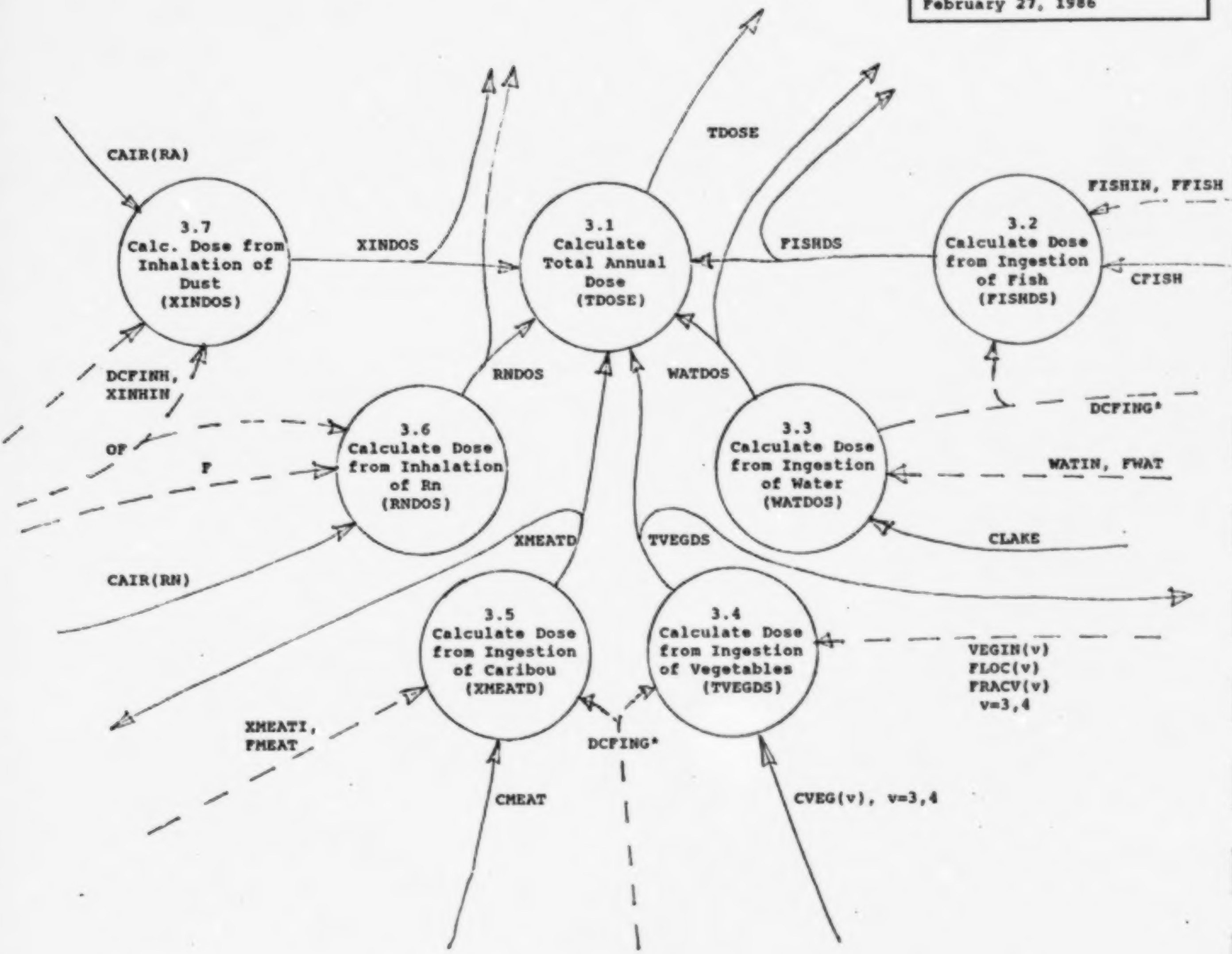
$$CMEAT = FIR * FBI * \sum_{v=1}^{2} FAUT(v) * CVEG(v) \quad [Bq \cdot g^{-1}] \quad (Eq.\ E.1.2)$$

v = 1,2 (lichen and grass only)

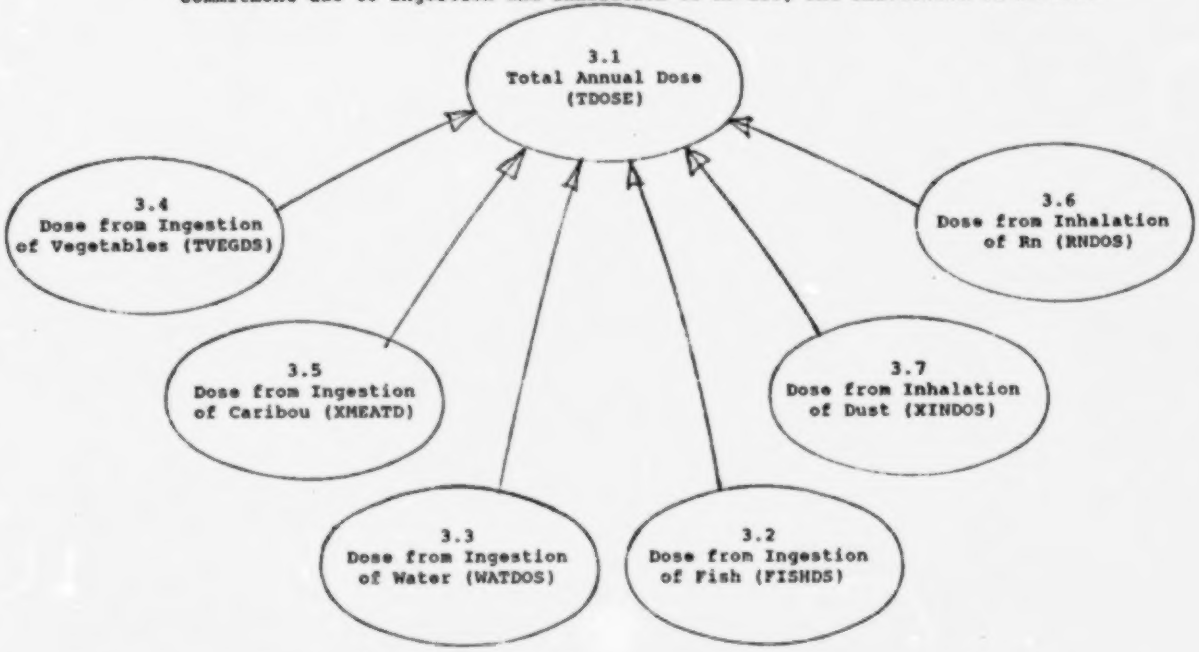
Unit check: $g\ a^{-1} * Bq\ g^{-1}\ Bq^{-1}\ a * (-) * Bq\ g^{-1} = [Bq\ g^{-1}]$

Figure 3.0
Calculate Dose

February 27, 1986


TDOSE
CAIR(RA)
3.7
Calc. Dose from
Inhalation of
Dust
(XINDOS)
XINDOS
3.1
Calculate
Total Annual
Dose
(TDOSE)
FISHDS
3.2
Calculate Dose
from Ingestion
of Fish
(FISHDS)
FISHIN, FFISH
CFISH
DCFINH,
XINHIN
RNDOS
WATDOS
3.6
Calculate Dose
from Inhalation
of Rn
(RNDOS)
OF
F
3.3
Calculate Dose
from Ingestion
of Water
(WATDOS)
DCFING*
WATIN, FWAT
CLAKE
CAIR(RN)
XMEATD
TVEGDS
3.5
Calculate Dose
from Ingestion
of Caribou
(XMEATD)
3.4
Calculate Dose
from Ingestion
of Vegetables
(TVEGDS)
VEGIN(v)
FLOC(v)
FRACV(v)
v=3,4
XMEATI,
FMEAT
DCFING*
CMEAT
CVEG(v), v=3,4

TDOSE TOTAL ANNUAL DOSE [Sv a^{-1}]. Total annual average effective whole body dose equivalent commitment due to ingestion and inhalation of Ra-226, and inhalation of Rn-222.


3.1
Total Annual Dose
(TDOSE)
3.4
Dose from Ingestion
of Vegetables (TVEGDS)
3.6
Dose from Inhalation
of Rn (RNDOS)
3.5
Dose from Ingestion
of Caribou (XMEATD)
3.7
Dose from Inhalation
of Dust (XINDOS)
3.3
Dose from Ingestion
of Water (WATDOS)
3.2
Dose from Ingestion
of Fish (FISHDS)

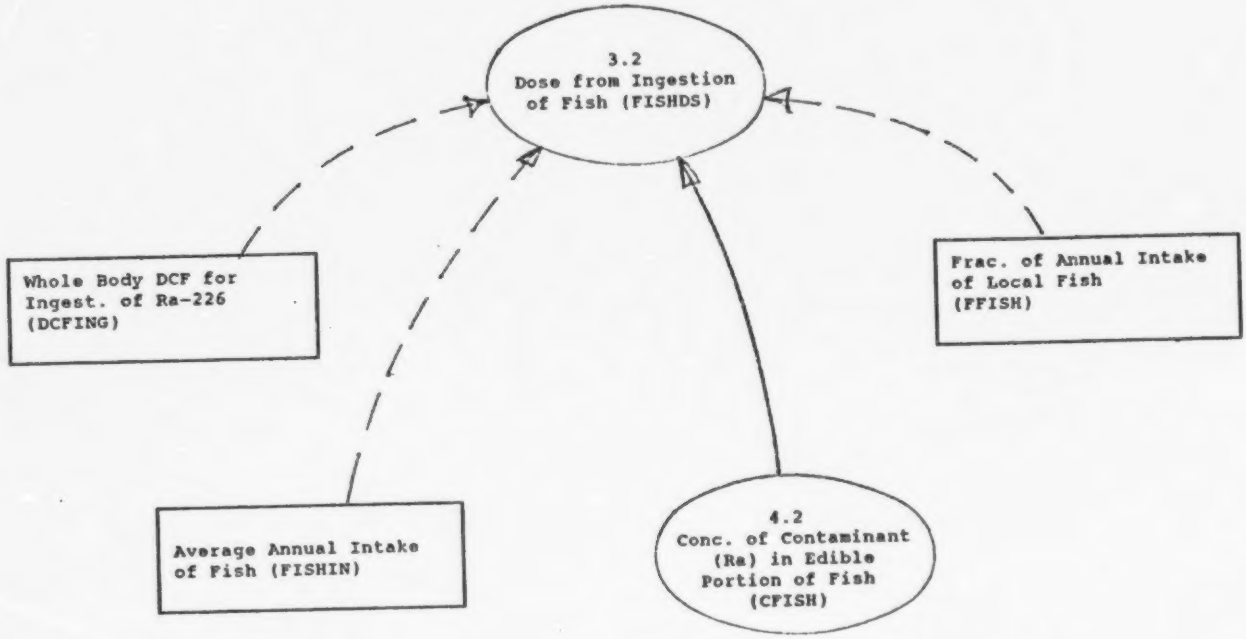
TDOSE = TVEGDS + XMEATD + WATDOS + FISHDS + XINDOS + RNDOS [Sv a^{-1}] (B. Goodwin)

Unit check: all variables are [Sv a^{-1}]

Figure 3.2
Calculate Dose
from Ingestion of Fish
(FISHDS)

March 4, 1986

FISHDS DOSE FROM INGESTION OF FISH [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to the ingestion of Ra-226 in fish meat.


3.2
Dose from Ingestion of Fish (FISHDS)
Whole Body DCF for Ingest. of Ra-226 (DCFING)
Average Annual Intake of Fish (FISHIN)
4.2
Conc. of Contaminant (Ra) in Edible Portion of Fish (CFISH)
Frac. of Annual Intake of Local Fish (FFISH)

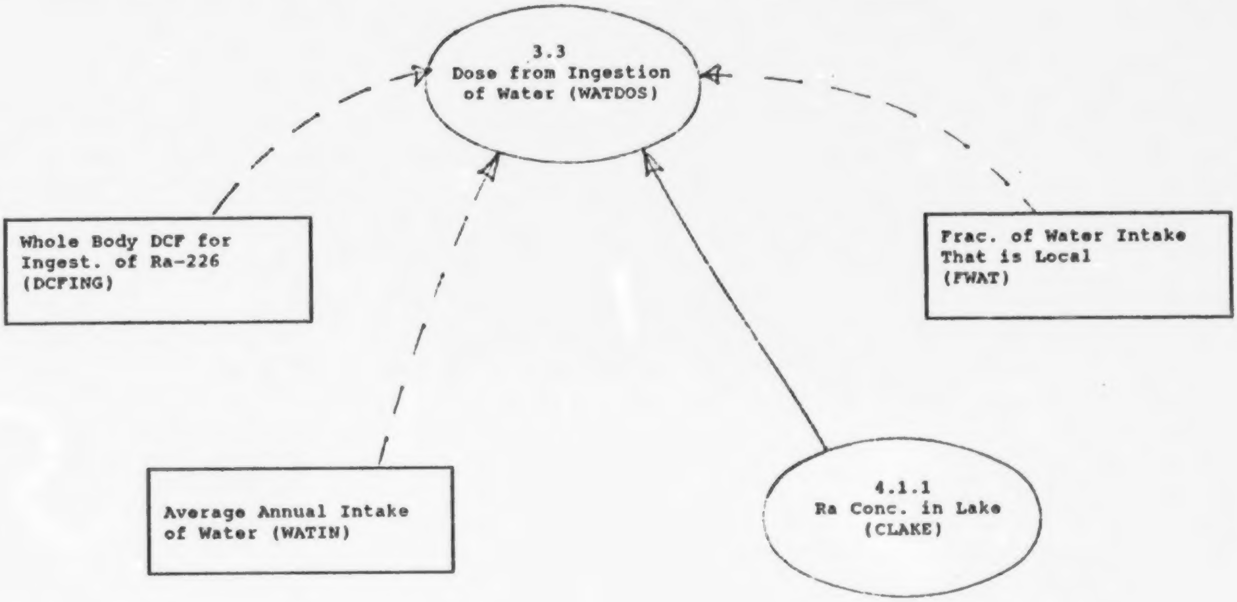
FISHDS = DCFING * FISHIN * CFISH * FFISH [Sv a^{-1}] (Eq. F.2.4)

Unit check: Sv Bq^{-1} * g a^{-1} * Bq g^{-1} * (-) = [Sv a^{-1}]

Figure 3.3
Calculate Dose
from Ingestion of
Water (WATDOS)

March 3, 1986

WATDOS DOSE FROM INGESTION OF WATER [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to the ingestion of Ra-226 in drinking water from a lake.

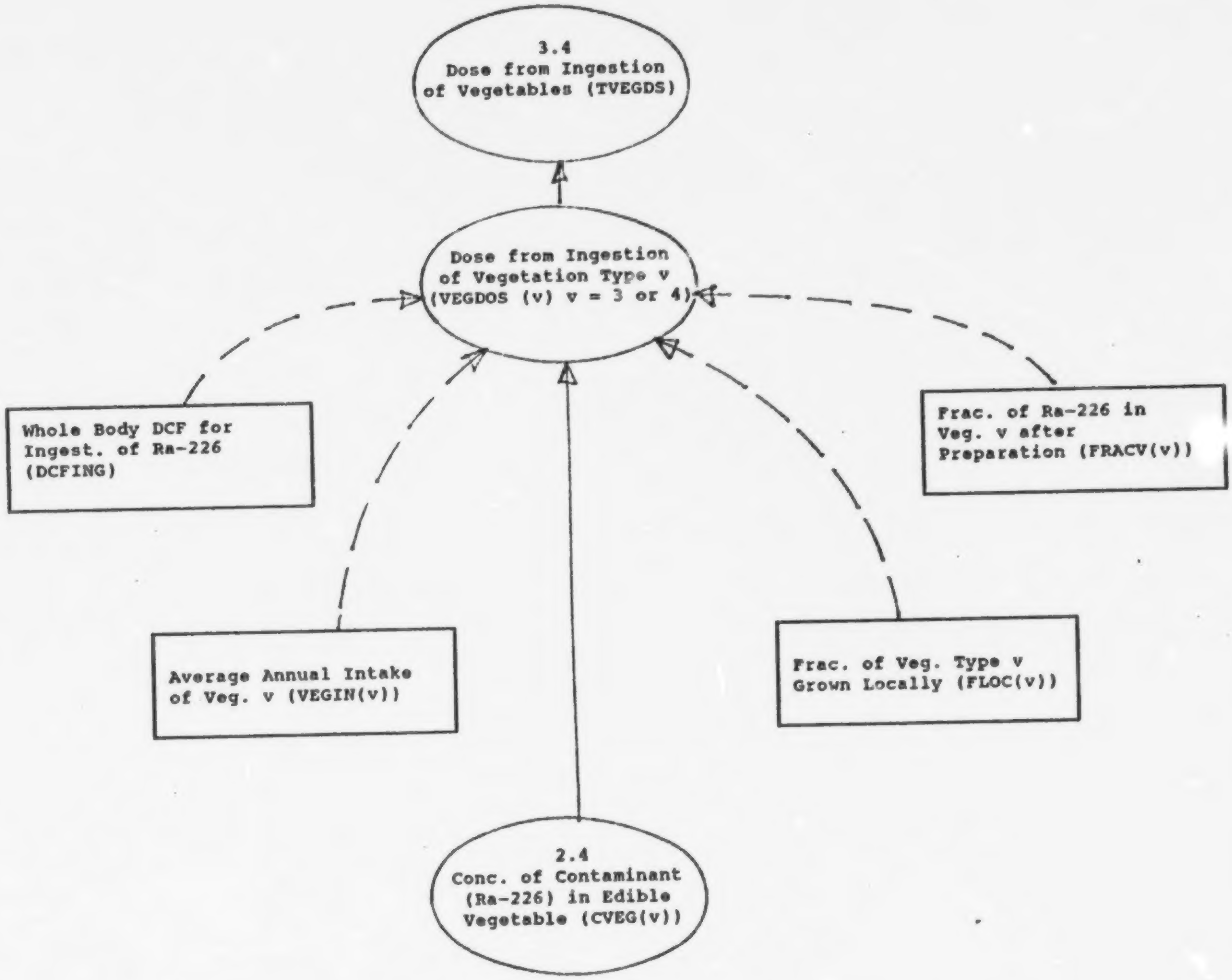

3.3
Dose from Ingestion
of Water (WATDOS)
Whole Body DCF for
Ingest. of Ra-226
(DCFING)
Frac. of Water Intake
That is Local
(FWAT)
Average Annual Intake
of Water (WATIN)
4.1.1
Ra Conc. in Lake
(CLAKE)

WATDOS = DCFING * WATIN * CLAKE * FWAT [Sv a^{-1}] (Eq. F.2.3)

Unit check: Sv Bq^{-1} * m^3 a^{-1} * Bq m^{-3} * (-) = [Sv a^{-1}]

Figure 3.4
Calculate Dose
from Ingestion of
Vegetables
(TVEGDS)
March 7, 1986

TVEGDS DOSE FROM INGESTION OF VEGETABLES [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to ingertion of Ra-226 in vegetables (above-ground and below-ground types only).


3.4
Dose from Ingestion
of Vegetables (TVEGDS)
Dose from Ingestion
of Vegetation Type v
(VEGDOS (v) v = 3 or 4)
Whole Body DCF for
Ingest. of Ra-226
(DCFING)
Frac. of Ra-226 in
Veg. v after
Preparation (FRACV(v))
Average Annual Intake
of Veg. v (VEGIN(v))
Frac. of Veg. Type v
Grown Locally (FLOC(v))
2.4
Conc. of Contaminant
(Ra-226) in Edible
Vegetable (CVEG(v))

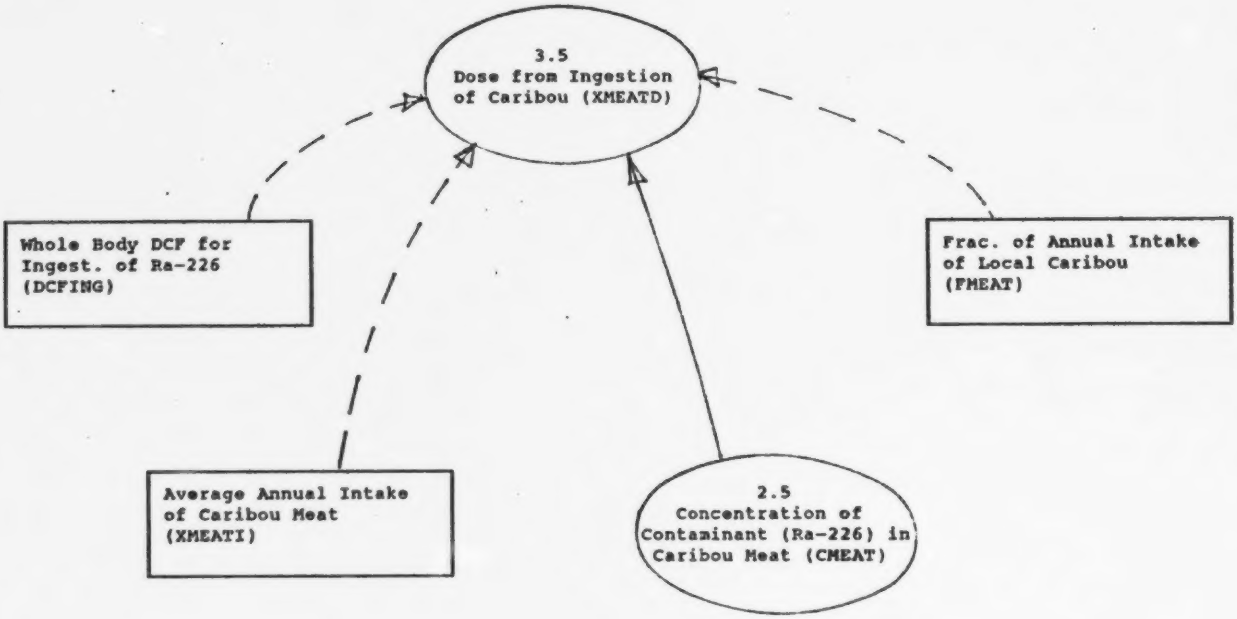
$$\mathrm{TVEGDS} = \sum_{v=3,4} \mathrm{VEGDOS}(v) \qquad [\mathrm{Sv\ a}^{-1}] \quad \text{(part of Eq. F.2.1)}$$

$$\mathrm{VEGDOS}(v) = \mathrm{DCFING} * \mathrm{VEGIN}(v) * \mathrm{CVEG}(v) * \mathrm{FRACV}(v) * \mathrm{FLOC}(v) \qquad [\mathrm{Sv\ a}^{-1}] \quad \text{(part of Eq. F.2.1)}$$

Unit check: VEGDOS: $\mathrm{Sv\ Bq}^{-1} * \mathrm{g\ a}^{-1} * \mathrm{Bq\ g}^{-1} * (-) * (-) = [\mathrm{Sv\ a}^{-1}]$

Note: v = 3,4 are vegetable types 3 and 4 from Figure 2.4. These are general above-ground and below-ground vegetables that are ingested by the 'critical' man.

XMEATD DOSE FROM INGESTION OF CARIBOU [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to the ingestion of Ra-226 in caribou meat.


3.5
Dose from Ingestion of Caribou (XMEATD)
Whole Body DCF for Ingest. of Ra-226 (DCFING)
Frac. of Annual Intake of Local Caribou (FMEAT)
Average Annual Intake of Caribou Meat (XMEATI)
2.5
Concentration of Contaminant (Ra-226) in Caribou Meat (CMEAT)

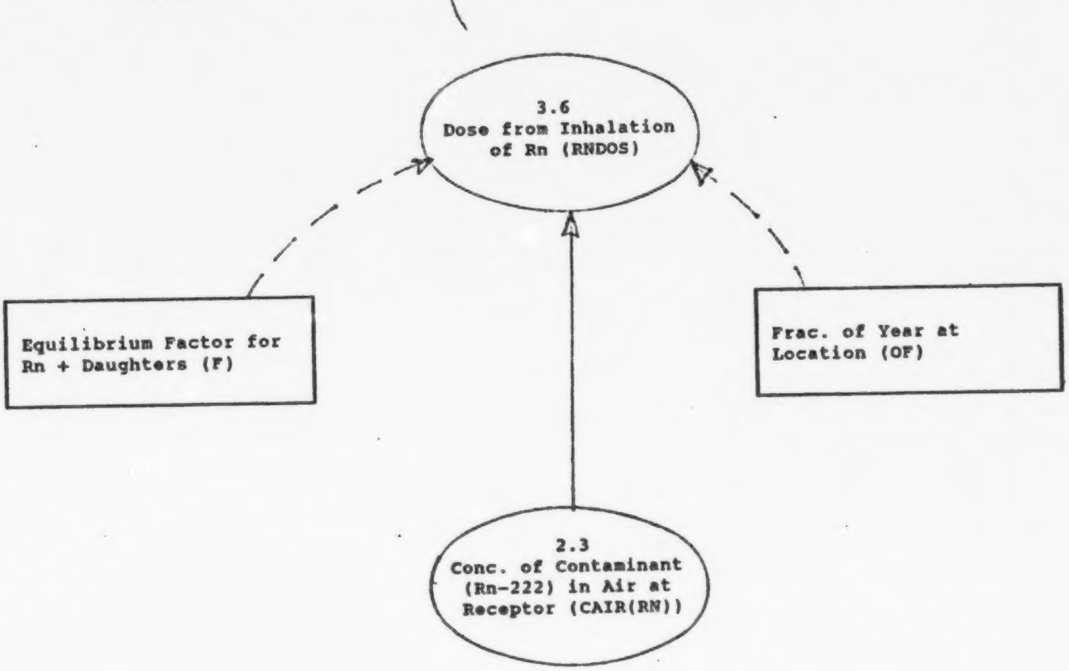
XMEATD = DCFING * XMEATI * CMEAT * FMEAT [Sv a^{-1}] (Eq. F.2.2)

Unit check: Sv Bq^{-1} * g a^{-1} * Bq g^{-1} * (-) = [Sv a^{-1}]

Figure 3.6
Calculate Dose
from Inhalation of Rn
(RNDOS)

March 3, 1986

RNDOS DOSE FROM INHALATION OF RN [$Sv\ a^{-1}$]. Annual average effective whole body dose equivalent commitment due to the inhalation of Rn-222 gas and its radioactive daughters.


3.6
Dose from Inhalation of Rn (RNDOS)
Equilibrium Factor for Rn + Daughters (F)
Frac. of Year at Location (OF)
2.3
Conc. of Contaminant (Rn-222) in Air at Receptor (CAIR(RN))

RNDOS = $1.09 \times 10^{-4} * F * OF * CAIR(RN)$ [$Sv\ a^{-1}$] (Eq. F.3.2)

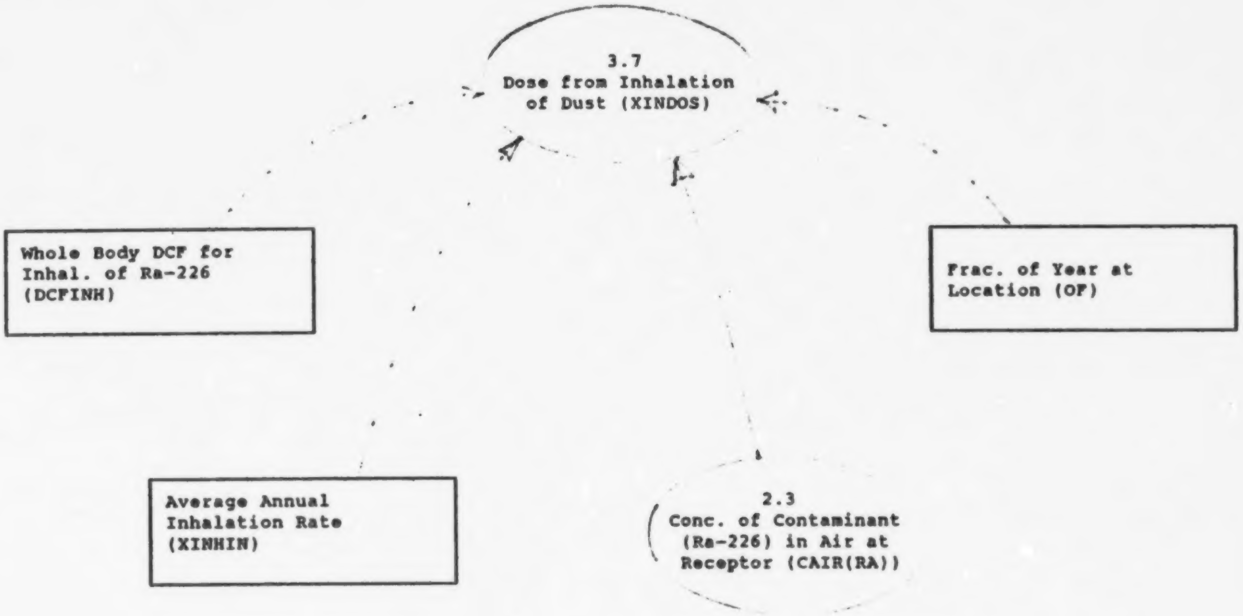
Unit check: $Sv\ Bq^{-1}\ m^{3}\ a^{-1} * (-) * (-) * Bq\ m^{-3} = [Sv\ a^{-1}]$

Note: The leading constant in the above equation has units of [$Sv\ Bq^{-1}\ m^{3}\ a^{-1}$].
See SENES Vol. 3, Appendix F, Section 3.2 for a description of this constant.

Figure 3.7
Calculate Dose from
Inhalation of Dust
(XINDOS)

March 3, 1986

XINDOS DOSE FROM INHALATION OF DUST [$\mathrm{Sv\ a^{-1}}$]. Annual average effective whole body dose equivalent commitment due to the inhalation of Ra-226 in dust.


3.7
Dose from Inhalation of Dust (XINDOS)
Whole Body DCF for Inhal. of Ra-226 (DCFINH)
Frac. of Year at Location (OF)
Average Annual Inhalation Rate (XINHIN)
2.3
Conc. of Contaminant (Ra-226) in Air at Receptor (CAIR(RA))

$$XINDOS = DCFINH * XINHIN * CAIR(RA) * OF \qquad [\mathrm{Sv\ a^{-1}}] \qquad \text{(Eq. F.3.1)}$$

Unit check: $\mathrm{Sv\ Bq^{-1} * m^{3}\ a^{-1} * Bq\ m^{-3}} * (-) = [\mathrm{Sv\ a^{-1}}]$

Figure 4.0
Calculate Concentrations
from Lake Pathways

March 4, 1986

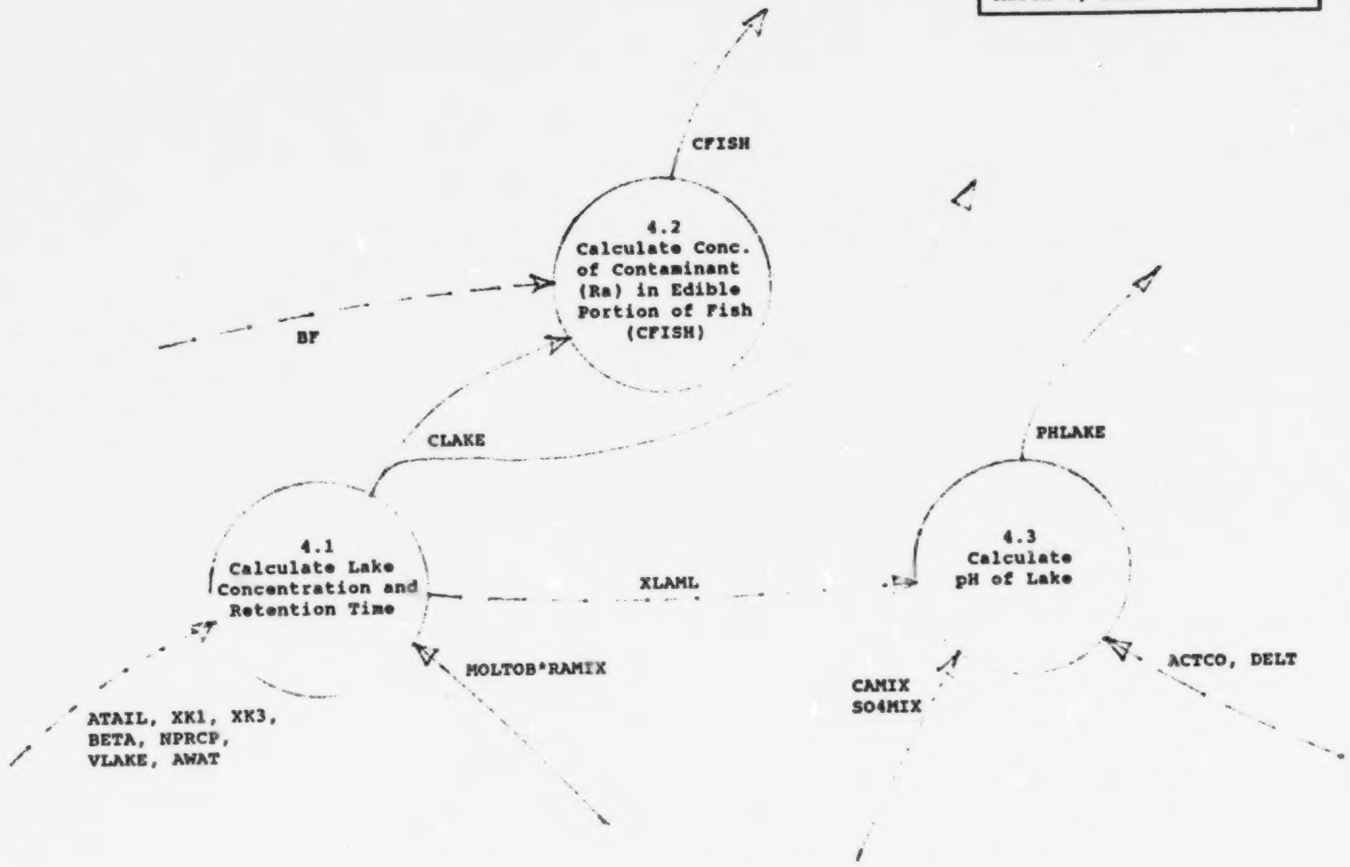

CFISH
4.2
Calculate Conc.
of Contaminant
(Ra) in Edible
Portion of Fish
(CFISH)
BF
CLAKE
PHLAKE
4.1
Calculate Lake
Concentration and
Retention Time
XLAML
4.3
Calculate
pH of Lake
MOLTOB*RAMIX
ACTCO, DELT
CAMIX
SO4MIX
ATAIL, XK1, XK3,
BETA, NPRCP,
VLAKE, AWAT

Figure 4.1
Calculate Lake
Concentration
and Retention Time

March 3, 1986

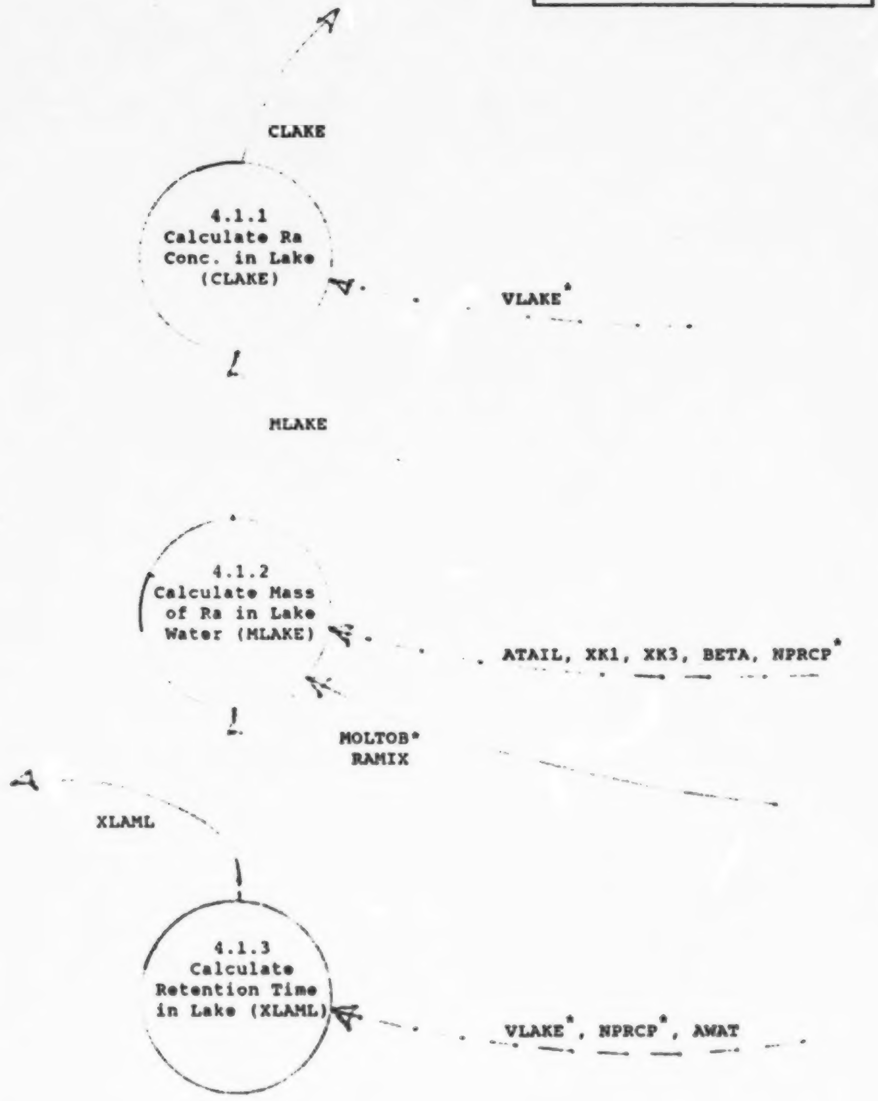
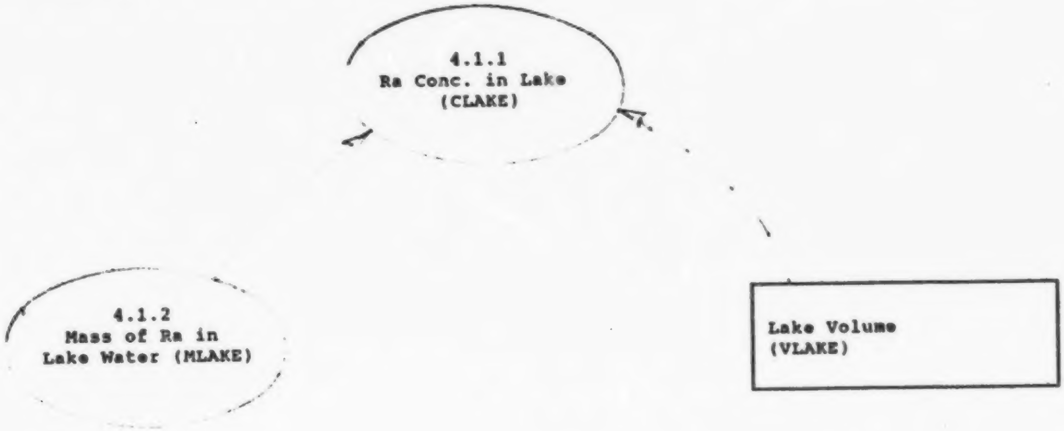

CLAKE
4.1.1
Calculate Ra
Conc. in Lake
(CLAKE)
VLAKE*
MLAKE
4.1.2
Calculate Mass
of Ra in Lake
Water (MLAKE)
ATAIL, XK1, XK3, BETA, NPRCP*
MOLTOB*
RAMIX
XLAML
4.1.3
Calculate
Retention Time
in Lake (XLAML)
VLAKE*, NPRCP*, AWAT

Figure 4.1.1
Calculate Ra Conc.
in Lake
(CLAKE)

March 5, 1986

CLAKE RA CONC. IN LAKE [Bq m^{-3}]. Average annual concentration of Ra-226 in the lake water as a function of time, where the source of Ra is runoff from the mixing zone.


4.1.1
Ra Conc. in Lake
(CLAKE)
4.1.2
Mass of Ra in
Lake Water (MLAKE)
Lake Volume
(VLAKE)

CLAKE = MLAKE/VLAKE [Bq m^{-3}] (page C-3)

Unit check: Bq/m^3 = [Bq m^{-3}]

Note: Symbols used in Appendix C:

CLAKE = $[Ra]_{L,t}$

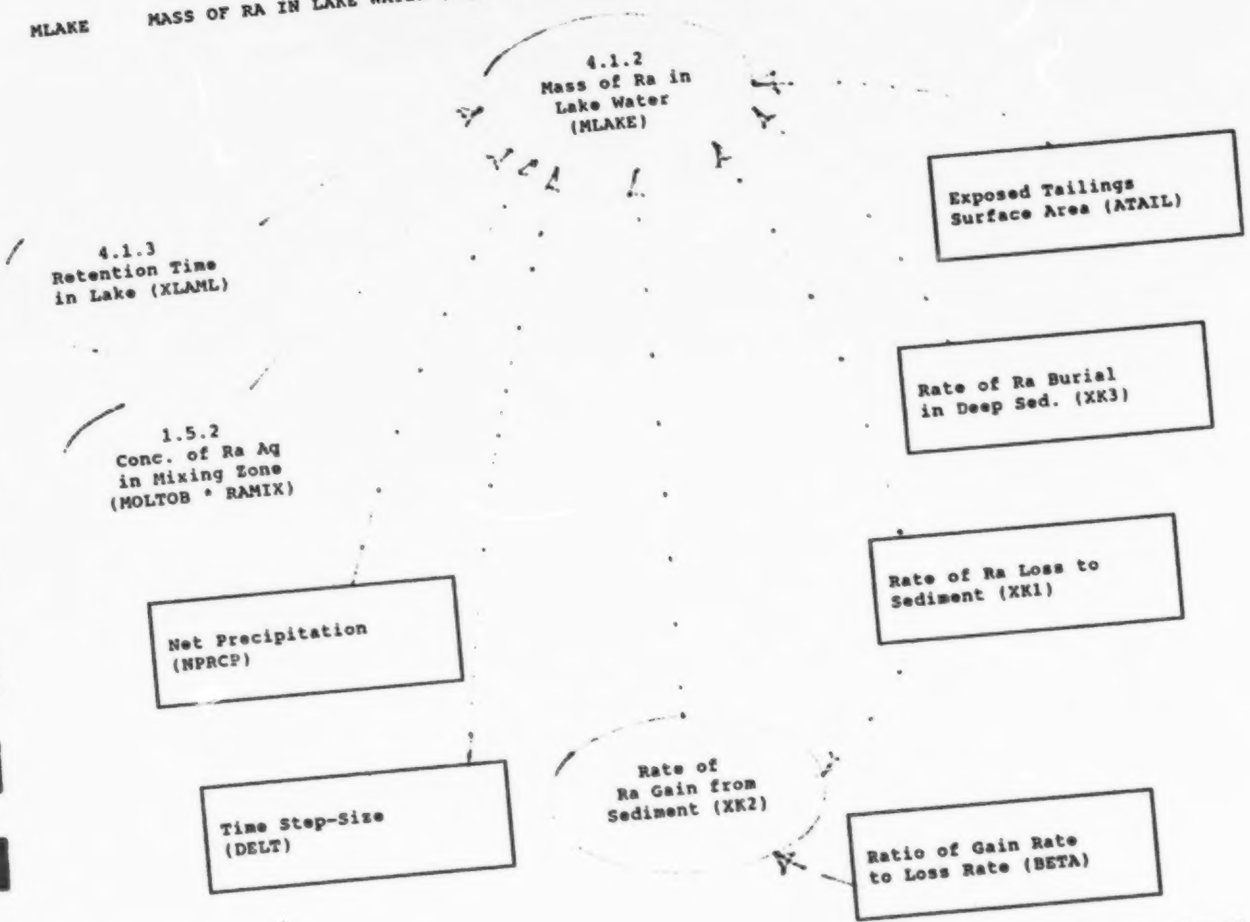
MLAKE = $[Ra]_{w,t}$ = $[Ra]_w$

VLAKE = V

Figure 4.1.2 (P. 1 of 2)
Calculate Mass of
Ra in Lake Water
(MLAKE)

April 11, 1986

MLAKE MASS OF RA IN LAKE WATER [Bq]. Radium activity in lake water as a function of time.


4.1.2
Mass of Ra in
Lake Water
(MLAKE)
Exposed Tailings
Surface Area (ATAIL)
4.1.3
Retention Time
in Lake (XLAML)
Rate of Ra Burial
in Deep Sed. (XK3)
1.5.2
Conc. of Ra Aq
in Mixing Zone
(MOLTOB * RAMIX)
Rate of Ra Loss to
Sediment (XK1)
Net Precipitation
(NPRCP)
Time Step-Size
(DELT)
Rate of
Ra Gain from
Sediment (XK2)
Ratio of Gain Rate
to Loss Rate (BETA)

(supplied on 10/2/86 in conversation with SENES)

$XK2 = XK1 * BETA$ $[a^{-1}]$

See below for MLAKE

Unit check: $XK2 = a^{-1} \times (-) = [a^{-1}]$

NOTES: 1. MLAKE and MSED are solved using a finite difference solution to a pair of coupled first-order linear differential equations, whose initial conditions are given as: (Page C-3)

$$MLAKE(0) = MSED(0) = 0$$

In a conversation with SENES on 10/4/86, it was noted that the initial conditions should be

$$MLAKE(-\Delta t) = MSED(-\Delta t) = 0$$

where Δt is the time step-size.

The finite-difference equations are given below, as specified by SENES.

2. No data dependency diagram is given for MSED, since MSED is never used, except in calculating MLAKE. If given, it would be identical to this one for MLAKE.

Figure 4.1.2
(Concluded)

3. Units of MLAKE and MSED are [Bq]. A unit check revealed that RAMIX must be converted from [mol m^{-3}] to [Bq m^{-3}] for use here. The conversion factor is MOLTOB.

4. Symbols used in Appendices A and C of SENES Vol. 3:

MLAKE = $[Ra]_{w,t} = [Ra]_w$
MSED = $[Ra]_{s,t} = [Ra]_s$
ATAIL = λ_T
XLAML = T_t
RAMIX = $[Ra]_{T,t} = [Ra]_{mix}$
DELT = Δt

XK1 = k_1
XK2 = k_2
XK3 = k_3
BETA = β
NPRCP = P

The equalities shown for RAMIX have been assumed, since $[Ra]_{T,t}$ and $[Ra]_{mix}$ have different definitions in Appendices A and C (B. Goodwin).

Equations and unit check for MLAKE:

MLAKE $(t + \Delta t) = K_w + A * \mathrm{EXP}(\lambda_1 \Delta t) + (X_t - A) * \mathrm{EXP}(\lambda_2 \Delta t)$ [Bq] (Eq. C.2.5)

where $K_w = (a_{13}\, a_{21} + a_{11}\, a_{23})$ / DENOM [Bq]

DENOM = $a_{12}\, a_{23} - a_{22}\, a_{13}$ $[a^{-2}]$

a_{11} = ATAIL * NPRCP * MOLTOB * RAMIX $\quad a_{21} = 0$ $[Bq\ a^{-1}]$

a_{12} = XK1 + 1/XLAML $\quad a_{22}$ = XK1 $[a^{-1}]$

a_{13} = XK2 $\quad a_{23}$ = XK2 + XK3 $[a^{-1}]$

$A = (b_1 + \lambda_1) * [(b_2 + \lambda_2)\, X_t - (b_1 + \lambda_2)\, Y_t] / [(b_2 - b_1) * b_3]$ [Bq]

$b_1 = a_{13} + a_{23} \quad b_2 = a_{12} + a_{22}$ $[a^{-1}]$

$b_3 = [(a_{12} + a_{23})^2 - 4 * \mathrm{DENOM}]^{\frac{1}{2}}$ $[a^{-1}]$

$\lambda_1 = (-a_{12} - a_{23} + b_3) / 2$ $[a^{-1}]$

$\lambda_2 = (-a_{12} - a_{23} - b_3) / 2$ $[a^{-1}]$

X_t = MLAKE $(t) - K_w$ [Bq]

Y_t = MSED $(t) - K_s$ [Bq]

MSED $(t + \Delta t) = K_s + B * \mathrm{EXP}(\lambda_1 \Delta t) + (Y_t - B) \times \mathrm{EXP}(\lambda_2 \Delta t)$ [Bq] (Eq. C.2.6)

$B = (b_2 + \lambda_1) * A / (b_1 + \lambda_1)$ [Bq]

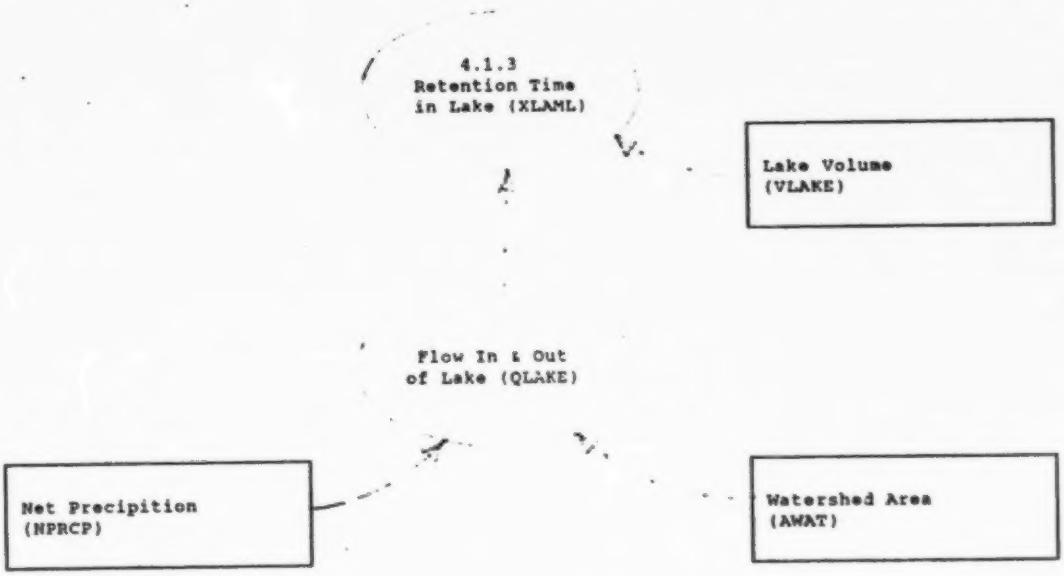
$K_s = (a_{12}\, a_{21} + a_{11}\, a_{22})$ / DENOM [Bq]

Unit check: MLAKE - [Bq], assuming a_{21} has the units shown.

Figure 4.1.3
Calculate Retention
Time in Lake
(XLAML)

March 5, 1986

XLAML RETENTION TIME IN LAKE [a]. Retention time (or travel time) in the reference lake.


4.1.3
Retention Time
in Lake (XLAML)
Lake Volume
(VLAKE)
Flow In & Out
of Lake (QLAKE)
Net Precipition
(NPRCP)
Watershed Area
(AWAT)

QLAKE = NPRCP * AWAT	$[m^3\ a^{-1}]$	(Eq. on p. C-6)
XLAML = VLAKE / QLAKE	[a]	(Eq. C.3.2)

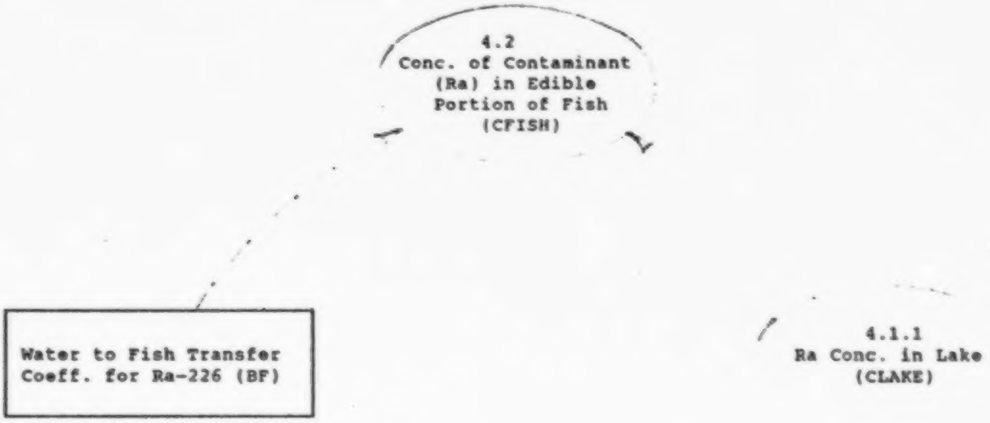
Unit check: $QLAKE = m\ a^{-1} * m^2 = [m^3\ a^{-1}]$
$XLAML = m^3 / [m^3\ a^{-1}] = [a]$

Note: Symbols used in Appendix C:

XLAML = T_t
VLAKE = V
QLAKE = Q
NPRCP = P
AWAT = A_W

Figure 4.2
Calculate Conc. of
Contaminant (Ra) in
Edible Portion of Fish
(CFISH)
March 3, 1986

CFISH CONC. OF CONTAMINANT (RA) IN EDIBLE PORTION OF FISH [Bq g^{-1} (wet weight)]. Average annual concentration of Ra-226 in edible portions of aquatic animals (fish).


4.2
Conc. of Contaminant (Ra) in Edible Portion of Fish (CFISH)
Water to Fish Transfer Coeff. for Ra-226 (BF)
4.1.1
Ra Conc. in Lake (CLAKE)

CFISH = BF * CLAKE [Bq g^{-1} (wet weight)] (Eq. on p. E.4)

Unit check: $Bq \cdot g^{-1} \cdot Bq^{-1} \cdot m^{3} * Bq\ m^{-3} = [Bq\ g^{-1}]$

Figure 4.3
Calculate
pH of Lake

March 5, 1986

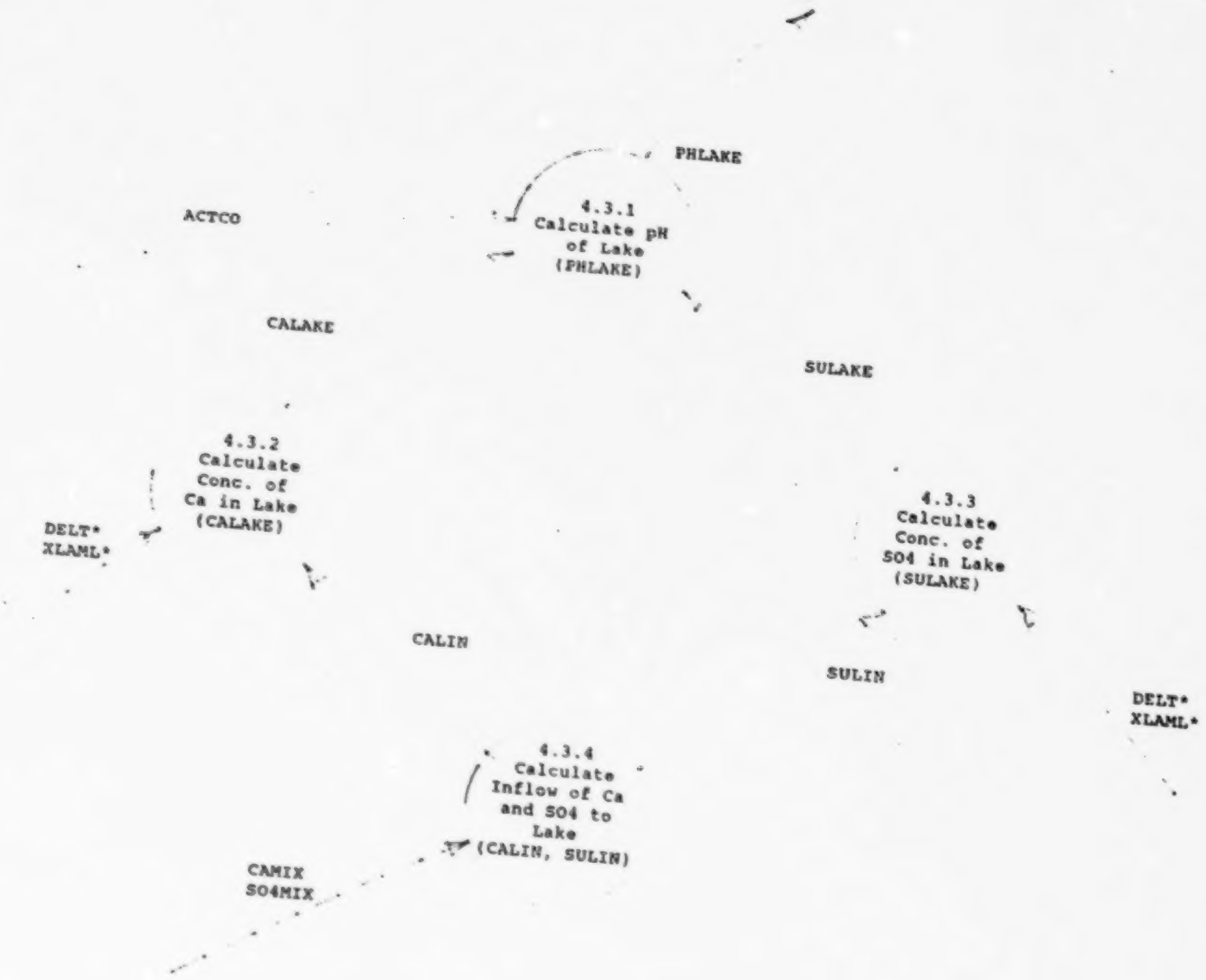
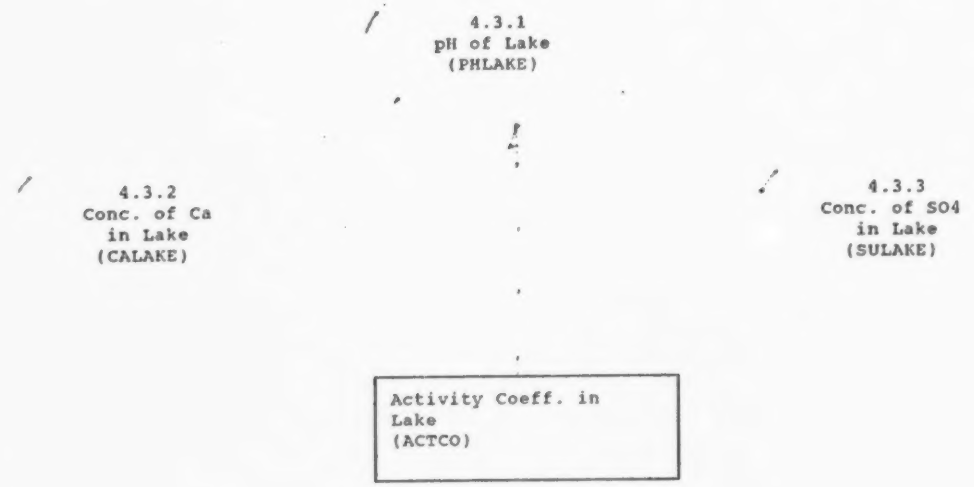

PHLAKE
ACTCO
4.3.1
Calculate pH
of Lake
(PHLAKE)
CALAKE
SULAKE
4.3.2
Calculate
Conc. of
Ca in Lake
(CALAKE)
DELT*
XLAML*
4.3.3
Calculate
Conc. of
SO4 in Lake
(SULAKE)
CALIN
SULIN
DELT*
XLAML*
4.3.4
Calculate
Inflow of Ca
and SO4 to
Lake
(CALIN, SULIN)
CAMIX
SO4MIX

Figure 4.3.1
Calculate pH
of Lake
(PHLAKE)

April 10, 1986

PHLAKE PH OF LAKE [-]. pH of the lake as a function of time.



Solve the following equation iteratively for pH (PHLAKE):

$$3\left[\frac{10^{8.5-3pH}}{(ACTCO)^{9/4}}\right] + 2\left[\frac{10^{3.91-2pH}}{ACTCO}\right] + \left[10^{0.3-pH}\right] + \left[10^{-3.86}\right]$$

$$+ 2 * CALAKE = SULAKE * \left[\frac{2-(ACTCO)^{3/4} * 10^{11.7-3pH}}{1+(ACTCO)^{3/4} * 10^{11.7-3pH}}\right] + \left[\frac{10^{pH-11.0}}{(ACTCO)^{1/4}}\right]$$

(Eq. on p. C-12 modified; See Note 1)

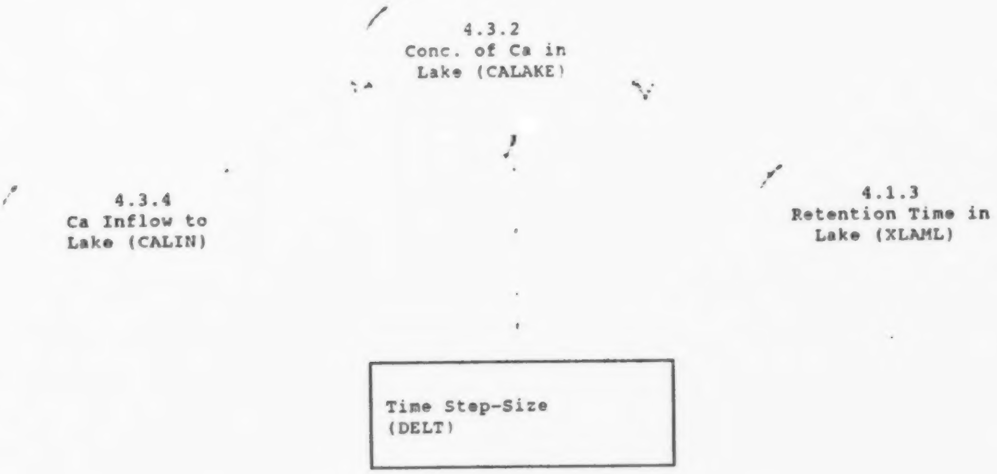
Initial guess: pH = 6 for the first time step and value from the previous time step for subsequent time steps (B. Goodwin).

Note:
1. Error noted by SENES on 10/2/86: the third term should be added (not subtracted).
2. Constants in the above equation are not given with units, so a unit check is not possible. The derivation of the equation has not been checked, since intermediate equations are not provided.
3. pH is used as an abbreviation for PHLAKE in the above equation.
4. CALAKE and SULAKE must be in units of mol/l to match the other terms in the above equation. The conversion factor is 10^{-3} l/m^3. CALAKE and SULAKE are to be premultiplied by 10^{-3} when used above.

Figure 4.3.2
Calculate Conc. of
Ca in Lake
(CALAKE)

April 11, 1986

CALAKE   CONC. OF CA IN LAKE [mol m^{-3}]. Concentration of calcium in the lake as a function of time.


4.3.2
Conc. of Ca in
Lake (CALAKE)
4.3.4
Ca Inflow to
Lake (CALIN)
4.1.3
Retention Time in
Lake (XLAML)
Time Step-Size
(DELT)

Solve the following finite difference equation for CALAKE:

$$\text{CALAKE}(t + \Delta t) = \text{CALAKE}(t) * \text{EXP}(-\text{DELT}/\text{XLAML}) + \text{CALIN}(t + \Delta t) * [1 - \text{EXP}(-\text{DELT}/\text{XLAML})]$$

[mol m^{-3}]   (See p. C-6, note 3)

Assume that $\text{CALAKE}(-\Delta t) = 10^{-1}$ [mol m^{-3}]   (See note 2)

Unit check: mol m^{-3} + mol m^{-3} * [EXP(a/a)] = [mol m^{-3}]

Notes: 1. Symbols used in Appendix C and above:

CALAKE = $[\text{Ca}]_{L,t}$
CALIN = $[\text{Ca}]_{in,t}$
XLAML = T_t
DELT = Δt
TIME = t

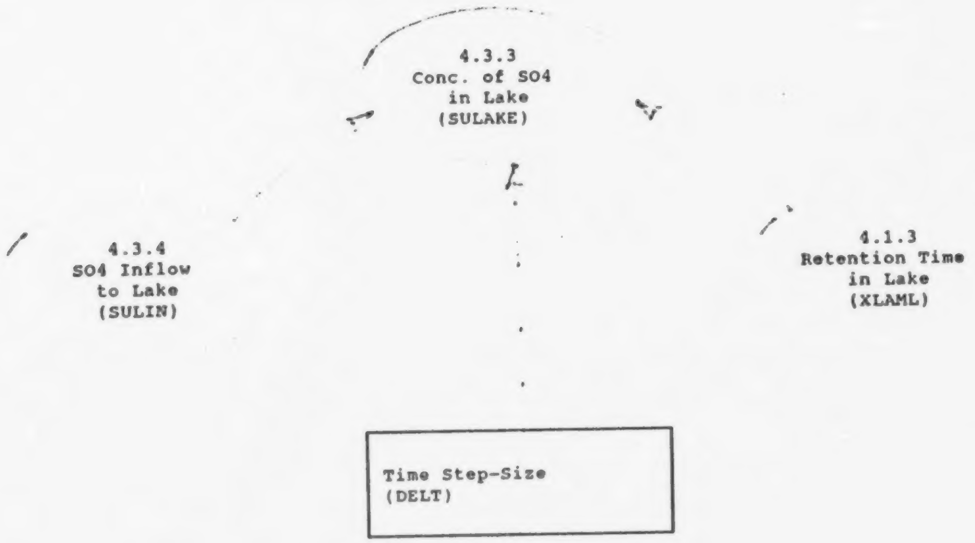
2. The initial condition is not given explicitly in SENES Vol. 3. It was originally assumed that the initial Ca concentration is at the "background" value (B. Goodwin). In a conversation with SENES on 10/4/86, it was noted that the initial condition given above was to be assumed.

3. The equation was changed according to corrections received from SENES on 9/4/86.

Figure 4.3.3
Calculate Conc.
of SO4 in Lake
(SULAKE)

April 11, 1986

SULAKE CONC. OF SO4 IN LAKE [mol m^{-3}]. Concentration of sulphate in the lake as a function of time.


4.3.3
Conc. of SO4
in Lake
(SULAKE)
4.3.4
SO4 Inflow
to Lake
(SULIN)
4.1.3
Retention Time
in Lake
(XLAML)
Time Step-Size
(DELT)

Solve the following finite difference equation for SULAKE:

SULAKE($t + \Delta t$) = SULAKE(t) * EXP(-DELT/XLAML) + SULIN($t + \Delta t$) * [1 - EXP(-DELT/XLAML)] [mol m^{-3}] (See P. C-6, note 3)

Assume that SULAKE($-\Delta t$) = $10^{-1.15}$ [mol m^{-3}] (See note 2)

Unit check: mol m^{-3} + mol m^{-3} * [EXP(a/a)] = [mol m^{-3}]

Notes: 1. Symbols used in Appendix C and above:

SULAKE = $[SO_4]_{L,t}$
SULIN = $[SO_4]_{in,t}$
XLAML = T_t
DELT = Δt
TIME = t

2. The initial condition is not given explicitly in SENES Vol. 3. It was originally assumed that the initial SO_4 concentration is at the "background" value (B. Goodwin). In a conversation with SENES on 10/4/86, it was noted that the initial condition given above was to be assumed.

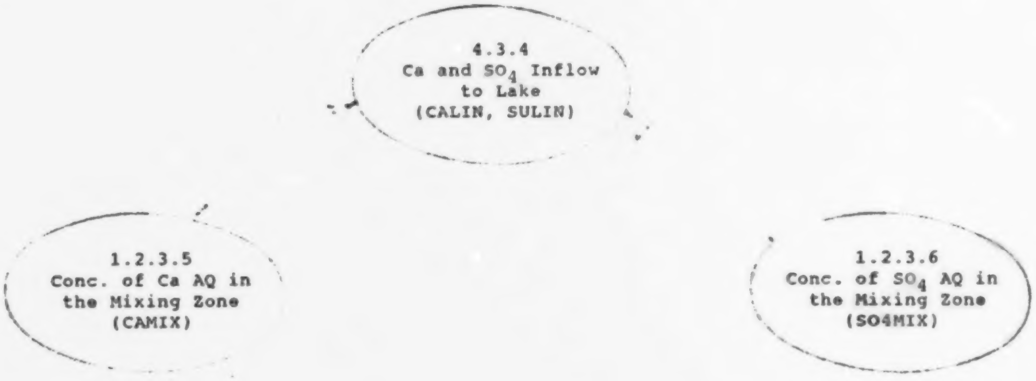
3. The equation was changed according to corrections received from SENES on April 9, 1986.

Figure 4.3.4
Calc. Inflow of Ca and SO_4 to Lake
(CALIN, SULIN)

May 12, 1986

CALIN CA INFLOW TO LAKE [mol m^{-3}]. Concentration of calcium in the aqueous phase entering the lake from the mixing zone and from background levels.

SULIN SO4 INFLOW TO LAKE [mol m^{-3}]. Concentration of sulphate in the aqueous phase entering the lake from the mixing zone and from background levels.


4.3.4
Ca and SO4 Inflow to Lake
(CALIN, SULIN)
1.2.3.5
Conc. of Ca AQ in the Mixing Zone
(CAMIX)
1.2.3.6
Conc. of SO4 AQ in the Mixing Zone
(SO4MIX)

$$CALIN = 2.5 \times 10^{-2} * CAMIX + 10^{-1} \quad [mol\ m^{-3}]$$ (P. C-5 modified, see Note 2)

$$SULIN = 2.5 \times 10^{-2} * SO4MIX + 10^{-1.15} \quad [mol\ m^{-3}]$$ (P. C-5 modified, see Note 2)

Unit check: $(-) * mol\ m^{-3} + mol\ m^{-3} = [mol\ m^{-3}]$

Note:
1. The first constant (2.5×10^{-2}) is a dilution factor, which is not given units, but which is assumed to be dimensionless.
2. The second constants (10^{-1} and $10^{-1.15}$) have been converted to [mol m^{-3}] using [mol m^{-3}] = [mol l^{-1}] x 10^{-3} [l m^{-3}]. Units are mixed up in the corresponding equations in SENES Vol. 3.
3. Symbols used in Appendices A and C:

   CALIN = $[Ca]_{in,t}$ CAMIX = $[Ca]_{T,t}$ = $[Ca^{2+}]_{mix}$
   SULIN = $[SO_4]_{in,t}$ SO4MIX = $[SO_4]_{T,t}$ = $[SO_4^{2+}]_{mix}$

   The equalities for CAMIX and SO4MIX are assumed since there are differences in the definition (and notations) of these variables in Appendices A and C.

Figure 5.0 Module Interface Specification for BISECT

P.1 BISECT

Find the root of a supplied function between two specified limits, using simple bisection (see R.W. Hamming. "Introduction to Applied Numerical Analysis", McGraw-Hill, 1971).

Original version 86-Mar-04 B.W. Goodwin
Current version 86-Mar-10 B.W. Goodwin

input only:

variable	type	dimension
lo-lim	numeric	scalar
hi-lim	numeric	scalar
accuracy	numeric	scalar
FUNC	function	

output only:

variable	type	dimension
root	numeric	scalar
error	numeric	scalar

argument descriptions:

lo-lim, hi-lim: inclusive limits within which must exist one and only one root.
accuracy: accuracy desired in locating a root. Must be greater than the machine precision for a double precision variable (ie [illegible] for our computers).
FUNC: external function to be searched. Calls are of the form: Z = FUNC (X), where X is the independent variable.
root: variable containing the root found.
error: error code which describes any problems that have been encountered.

Note: low-lim, hi-lim, accuracy and root are double precision variables. FUNC is a double precision function. The error code can be an integer variable with pre-defined values.

method:

At least one root exists if FUNC(lo-lim) and FUNC(hi-lim) have opposite signs. For this case, keep bisecting until the range is less than or equal to the specified accuracy, or until a root is found. A root has been found if a reference to FUNC returns a value of zero. When the range has been reduced, a root exists somewhere in the

Figure 5.0
(continued)

P.2 **BISECT**

range if FUNC still has opposite signs at the two limits of the range. If FUNC(lo-lim) has the same sign as FUNC(hi-lim), then there are zero or an even number of roots over the specified range - an error condition. If FUNC has the same sign at the two limits of the range, then a pole has been found - another error condition.

sample tests:

The code should work for the following sample functions and limits:

function	limits	root
X**2 - 2	[illegible]	2**(1/2)
X**3 - 2	[1, [illegible]]	2**(1/3)
(X-2)*(X-3)	[illegible]	(error)
(X-2)*(X-3)	[[illegible].1, [illegible]]	3
(X-2)*(X-3)	[[illegible],2.[illegible]]	2
(X-2)*(X-3)*(X-4)	[3.1,[illegible]]	4
(X-2)	[3, 4]	(error)
(X-2)	[2, [illegible]]	2
(X-2)	[3, 3]	(error)

APPENDIX C: DATA DICTIONARY FOR THE TAILINGS MODEL

Specifications for the UTAP (Uranium Tailings Assessment Program) model have been developed from the SENES Phase II Report.[1] The specifications consist of three components: data flow diagrams (DFD's), minispecifications (minispecs), and the data dictionary (DD). The following notes describe the format of the data dictionary, and is followed by the data dictionary developed from the uranium tailings model. Notes for the DFD's and minispecs are presented in Appendix B.

Data dictionary

The data dictionary is an alphabetical list of all variables and key words used in the specifications. The following fields are presented for each variable:

- Short name: Every variable has a short name of at most six characters that can be used as a FORTRAN variable name. This appears first and is used for alphabetical ordering of the data dictionary.
- Long name: Every variable also has a long name consisting of several words, often abbreviated, to be used in formal references to the variable, such as in the data flow diagrams and minispecs.
- SI units: Metric units are provided for every variable. They appear inside square brackets (e.g. [mol s^{-1}]). Where no units apply, "[-]" is used.
- Definition: Each variable is defined in one or two sentences that explain what physical quantity the variable represents. The definition states if the variable depends on time or space variables, or is specific to one nuclide. It contains all the relevant information from SENES documentation.
- Background information: The data dictionary states where every variable was generated and used in the SENES code, what problems have arisen in using the documentation related to that variable, and what consequent corrections have been made.

---

1)SENES, 1985. Probabilistic model development for the assessment of the long-term effects of uranium mill tailings in Canada - Phase II; Volume 3 - Probabilistic analysis user's manual and documentation for the Uranium Tailings Assessment Program (UTAP); a research report prepared under contract # OSQ84-00207 for The National Uranium Tailings Program, Canada Centre for Mineral and Energy Technology, Energy, Mines and Resources Canada, 555 Booth Street, Ottawa, Ontario, K1A OG1.

- Sampled/calculated: The data dictionary states whether a variable is an input (sampled) parameter or a calculated parameter. If the variable is a sampled parameter, the input data (including probability distribution data) is listed. If the variable is calculated, the appropriate equation is referenced.

- Equation: Where a variable is calculated according to a simple equation, that equation also appears in the data dictionary.

In addition to definitions of variables, the data dictionary also contains definitions of key words like "MODULE" and "UNSAT. ZONE" that appear frequently and have a specific meaning.

DATA DICTIONARY FOR THE NUTP TAILINGS SUBMODEL
(1986-Mar-04)

ACTCO	ACTIVITY COEFFICIENT IN LAKE [-]. Activity coefficient for divalent ions used in the lake pH calculations, where the ionic strength is relatively low. Used in the LAKEPH modules. ACTCO is sampled from the following distribution: Constant (0.8) (as corrected by SENES on 10/2/86 and in a letter dated 13-Feb-1986).
AEQBM	SQ. ROOT OF EQBM CONST. [mol m^{-3}]. AEQBM is an intermediate variable, equal to the square root of the equilibrium constant for the dissolution solubility of gypsum, divided by the activity coefficient. Intermediate variable calculated in and used throughout the TSOURC modules. AEQBM is calculated using an expression near equation (3.20) in Appendix A of SENES Vol. 3: $AEQBM = (KSUBSO)^{1/2} / FACTCO$
AEXPT	EXPOSED TAILINGS SURFACE AREA [m^2]. Exposed surface area of the tailings, nominally as a function of time but assumed to be constant in this version of the tailings submodel. AEXPT and ATAIL appear to be the same parameter with different names, and used interchangeably. Both AEXPT and ATAIL are sampled (input) parameters used in the DUST, RADON and LAKE modules. Both are sampled from: Constant (5×10^5)
AIRDIS	MODULE. All processes required to calculate concentrations in air at the receptor, given source concentrations at the tailings source (see DFD 2 and DDD 2.3). The major outputs from AIRDIS are CAIR(RA) and CAIR(RN).
ATAIL	EXPOSED TAILINGS SURFACE AREA [m^2]. Exposed surface area of the tailings, nominally as a function of time but assumed to be constant in this version of the tailings submodel. AEXPT and ATAIL appear to be the same parameter with different names, and used interchangeably. Both AEXPT and ATAIL are sampled (input) parameters used in the DUST, RADON and LAKE modules. Both are sampled from: Constant (5×10^5)
AWAT	WATERSHED AREA [m^2]. Watershed area feeding the lake. Input parameter used in the LAKE module and sampled from: Constant ($2X10^7$)
BETA	RATIO OF GAIN RATE TO LOSS RATE [-]. Parameter relating the rate constant XK1 to rate constant XK2. Input parameter used in the LAKE module and sampled from: Lognormal (0.18, 1.35, 0.1, 0.328) (as corrected by SENES on 10/2/14 and with a modification to the last parameter in a letter dated 13-Feb.-1986).
BF	WATER TO FISH TRANSFER COEFF. FOR RA-226 [Bq g^{-1} (wet weight) x Bq^{-1} m^3]. Water to aquatic produce (fish) transfer coefficient (Ra-226 to fish flesh). Input parameter used in the FISH module and sampled from: Lognormal ($1.6x10^{-5}$, 3.5, $1.0x10^{-6}$, $6.0x10^{-4}$)

CAIR(RA)	CONC. OF CONTAMINANT (RA-226) IN AIR AT RECPTOR [Bq m^{-3}]. Average annual concentration of Ra-226 associated with dust in air at a receptor (locally grown vegetables and/or a 'critical' man). Variable calculated in the AIRDIS module and used in the DOSE and VEGDEP modules. Calculated using the equation in Appendix D of SENES Vol. 3.
CAIR(RN)	CONC. OF CONTAMINANT (RN-222) IN AIR AT RECPTOR [Bq m^{-3}]. Average annual concentration of Rn-222 gas in air at a receptor (a 'critical' man). Variable calculated in the AIRDIS module and used in the DOSE module. Calculated using the equation in Appendix D of SENES Vol. 3.
CALAKE	CONC OF CA IN LAKE [mol m^{-3}]. Concentration of calcium in the lake as a function of time. Intermediate variable calculated in the LAKEPH module. Calculated from a finite difference equation near equation (3.2) in Appendix C of SENES VOL. 3, with the unstated assumption that $CALAKE = 0$ at $TIME = 0$. Note: to keep consistency with other units, CALIN must be used in [mol m^{-3}] when this finite difference equation is used.
CALIN	CA INFLOW TO LAKE [mol m^{-3}]. Concentration of calcium in the aqueous phase entering the lake from the mixing zone and from background levels. Intermediate variable calculated in the LAKEPH module. Calculated using an equation before equation (3.1) in Appendix C of SENES Vol. 3: $CALIN = 2.5 \times 10^{-2} \times CAMIX + 10^{-1}$ Note 1: the first constant in the above equation is a dilution factor, and the second is the "background" level of Ca concentrations; see SENES Vol. 3, p C-5. Note 2: to keep consistency with other variables, CALIN must be used in [mol m^{-3}] when used to calculate CALAKE. This requires the conversion of background calcium concentrations from [mol l^{-1}] to [mol m^{-3}], which has been done in the above equation.
CAMIX	CONC. OF CA AQ IN THE MIXING ZONE [mol m^{-3}]. Concentration of Ca in the aqueous phase of the mixing zone. Variable calculated in the TSOURC modules, and used in the LAKEPH module. Calculated using equation (7.1) in Appendix A of SENES Vol. 3, and other equations nearby: $CAMIX = CONCCA \quad (TIME <= TFINAL)$ $CAMIX = ION \quad (TIME > TFINAL \text{ and } GYPLFT > 0)$ $CAMIX = 0 \quad (TIME > TFINAL \text{ and } GYPLFT <= 0)$
CFISH	CONC. OF CONTAMINANT (RA) IN EDIBLE PORTION OF FISH [Bq g^{-1} (wet weight)]. Average annual concentration of Ra-226 in edible portions of aquatic animals (fish). Variable calculated in the FISH module and used in the DOSE module. Calculated from an equation in Appendix E (Section 2.1) in SENES Vol. 3: $CFISH = BF * CLAKE$
CHMCNS	CHEMICAL CONSTANTS is a list of three variables that are mostly used together. It consists of AEQBM, CSINIT, and KRATE.
CLAKE	RA CONC. IN LAKE [Bq m^{-3}]. Average annual concentration of Ra-226 in the lake water as a function of time, where the source of Ra is runoff from the mixing zone. Variable calculated in the LAKE module and used in the DOSE and FISH modules. Calculated using an equation near equation (2.4) in Appendix C of SENES Vol. 3: $CLAKE = MLAKE / VLAKE$

CMEAT	CONCENTRATION OF CONTAMINANT (RA-226) IN CARIBOU MEAT [Bq g^{-1}(wet weight)]. Average annual concentration of Ra-226 in caribou meat, due to the ingestion of lichen and grass (vegetation types 1 and 2) which have been contaminated by air deposition. Variable calculated in the VEGEAT module and used in the DOSE module. Calculated using equation (1.2) in Appendix E of SENES Vol. 3.
CONCCA	CONC. OF CA AQ. IN THE UNSAT. ZONE [mol m^{-3}]. Concentration of calcium ions in the aqueous phase of the unsaturated zone, as a function of time and position. Intermediate variable calculated in the TSOURC modules. Calculated using equation (3.9) in Appendix A of SENES Vol. 3.
CONCRA	INIT. SOLID PHASE ACT. OF RA-226 IN TAILINGS [Bq g^{-1}]. Initial solid phase activity of Ra-226 in the tailings. Input parameter used in the TSOURC modules and sampled from: Triangular (5.0, 11.0, 15.0)
CONCS8	CONC. OF SO4 AQ IN THE UNSAT. ZONE [mol m^{-3}]. Concentration of sulphate ions in the aqueous phase of the unsaturated zone, due to pyrite oxidation and downward movement of water, as a function of time and position. Intermediate variable calculated in the TSOURC modules. Calculated using equation (3.16) in Appendix A of SENES Vol. 3. Note that most uses of CONCS8 appear to be at position z=DUNSAT, with a corresponding simplification to equation (3.16). The only known (possible) exception is where PHUNST is required as a function of depth.
CRAAVG	AVG. CONC. OF RA TO MIXING ZONE [mol m^{-3}]. Average concentration of Ra-226 in the aqueous phase leaving the bottom of the unsaturated zone and entering the mixing zone. Intermediate variable calculated in the TSOURC modules. Calculated using equation (3.25) in Appendix A of SENES Vol. 3 and using another equation not given in the SENES report: CRAAVG = RAFTH (TIME >=TFINAL) CRAAVG = RAFTH + RAAQ (TIME < TFINAL)
CRAGIN	INIT. CONC. OF RA IN GYPSUM [mol m^{-3}]. Initial concentration of Ra-226 associated with gypsum in the tailings. This is a bulk concentration. Intermediate variable calculated in the TSOURC modules. Calculated from an expression near equation (2.2) in Appendix A of SENES Vol. 3: CRAGIN = RAINGP x RADIUM
CSINIT	INIT. CONC. OF SULPHATE AT BOTTOM OF UNSAT. ZONE [mol m^{-3}]. Initial concentration of sulphate ions in the aqueous phase at the bottom of the unsaturated zone (ie. at the depth DUNSAT). Intermediate variable calculated in and used throughout the TSOURC modules, using an expression near equation (3.15) in Appendix A of SENES Vol. 3: CSINIT = SFLUX / NPRCP
CTH230	INIT. SOLID PHASE CONC. OF TH-230 [mol m^{-3}]. Initial solid phase concentration of Th-230 in the tailings. Intermediate variable calculated from equation (2.3) of Appendix A of SENES Vol. 3: CTH230 = RADIUM * RALAM / THLAM
CTH232	INIT. SOLID PHASE CONC. OF TH-232 [mol m^{-3}]. Initial solid phase concentration of Th-232 in the tailings. Intermediate variable calculated from equation (2.4) of Appendix A of SENES Vol. 3: CTH232 = UO2TON x RHOTAL x 3.788×10^{-6} / UPERTH

CVEG(v)	CONC. OF CONTAMINANT (RA-226) IN EDIBLE VEGETABLE [Bq g^{-1}(wet weight)]. Average annual concentration of Ra-226 in edible vegetables due to foliar retention of Ra-226 deposited from dust in the air. Four types of vegetation are considered (v=1,2,3,4 =lichen, grass, above-ground and below-ground classes). Variable calculated in the VEGDEP module and used in the VEGEAT (for v=1,2) and in the DOSE (for v=3,4) modules. Calculated using equation (1.1) in Appendix E of SENES Vol. 3.
DCFING	WHOLE BODY DCF FOR INGEST. OF RA-226 [Sv Bq^{-1}]. Whole body dose conversion factor for the ingestion of Ra-226. Input parameter used in the DOSE module and sampled from: Constant (3.2×10^{-7})
DCFINH	WHOLE BODY DCF FOR INHAL. OF RA-226 [Sv Bq^{-1}]. Whole body dose conversion factor for the inhalation of Ra-226. Input parameter used in the DOSE module and sampled from: Constant (2.6×10^{-6})
DELH	INCREMENTAL CHANGE IN DEPTH1 FROM TIME TO TIME+DELT [m]. Incremental change in the depth of the upper (gypsum-depleted) layer of the unsaturated zone from time = TIME to time = TIME+DELT. Intermediate variable calculated in the TSOURC modules. Calculated from the following equations (not given in SENES Vol. 3): DELH = DEPTH1(at TIME+DELT) - DEPTH1(at TIME) (for TIME < TFINAL) DELH = 0 (FOR TIME >=TFINAL) Note 1: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone. Note 2: DELH is an intrinsically non-negative variable, with a value of zero at TIME=0 and TIME=TFINAL, and with positive values in between.
DELT	TIME STEP-SIZE [a]. Time step-size used within various equations, such as the finite difference equation for CALAKE. Input parameter used in the TSOURC and LAKE modules. Not specified in the Appendices, but assigned a value of 10 [a] in Table 5.1 of SENES Vol. 3.
DEPTH1	DEPTH OF UPPER UNSAT. ZONE [m]. Depth of the upper (gypsum-depleted) layer of the unsaturated zone. Variable calculated in the TSOURC modules, and used throughout TSOURC and in the RADON module. Calculated from an iterative equation, equation (2.30) in Appendix A of SENES Vol. 3. Note: equation (2.30) applies only for TIME < TFINAL. For TIME >= TFINAL, use DEPTH1 = DUNSAT (TIME >= TFINAL) Note 1: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone. Note 2: DEPTH1 is an intrinsically non-negative variable, ranging from zero at TIME=0 to DUNSAT at TIME=TFINAL.
DEPTH2	DEPTH OF LOWER UNSAT. ZONE [m]. Depth of the lower (gypsum-enriched) layer of the unsaturated zone. Variable calculated in the TSOURC modules, and used in the RADON module. Calculated using the following equations (which are not given in SENES Vol. 3): DEPTH2 = DUNSAT - DEPTH1 (TIME < TFINAL) DEPTH2 = 0 (TIME >=TFINAL)
DEPVEL	DEPOSITION VELOCITY [m s^{-1}]. Deposition velocity for Ra in dust particulates. Input parameter used in the VEGDEP module, and sampled from: Uniform (0.003, 0.03)

DIFLEN	RECIPROCAL OF RN DIFFUSION PATH LENGTH [m^{-1}]. Reciprocal of the Rn-222 diffusion path length. Intermediate variable calculated in the RADON module. Calculated using an equation near equation (2.2) in Appendix B of SENES Vol. 3: $DIFLEN = [RNLAM \times TVOID / EBKDFC]^{0.5}$
DISTCO	RA SOLID/AQ DIST. COEFFICIENT [-]. Radium solid/liquid distribution coefficient in the unsaturated zone. Input parameter used in the TSOURC modules and sampled from: Uniform (0.01, 0.1)
DMIX	DEPTH OF MIXING ZONE [m]. Depth of the mixing zone, which is part of the saturated zone underlying the unsaturated zone. It is assumed that the source of contaminated surface waters derives from releases from the mixing zone. Input parameter used in the TSOURC modules and sampled from: Constant (1.0)
DOSE	MODULE. All processes required to calculate the dose, given relevant concentrations in various environmental pathways (see DFD 3 and DDD 3.1 to 3.7). The major outputs from DOSE are TDOSE, TVEGDS, XMEATD, WATDOS, FISHDS, XINDOS, and RNDOS.
DUNSAT	DEPTH OF UNSAT. ZONE [m]. Depth of the unsaturated zone, including the upper (gypsum-depleted) and lower (gypsum-enriched) layers. Input parameter used in the TSOURC modules and sampled from: Constant (1.0) Note: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone.
DUREXP(v)	DURATION OF EXPOSURE OF VEGETATION [s]. Duration of exposure of vegetable types v (v=1,2,3,4 = lichen, grass, above- ground, below-ground) to deposition. Input parameter used in the VEGDEP module and sampled from: Uniform (9.0×10^7, 1.1×10^8) (v=1) Uniform (2.7×10^6, 3.1×10^6) (v=2) Uniform (4.7×10^6, 5.6×10^6) (v=3,4)
DUST	MODULE. All processes required to calculate the emission rate of Ra-226 associated with dust at the tailings surface (see DFD 2 and DDD 2.1). The major output from DUST is EMRATE(RA).
EBKDFC	EFFECTIVE BULK DIFFUSION COEFFICIENT [$m^2\ s^{-1}$]. Effective bulk diffusion coefficient of Rn-222 in the unsaturated zone of the tailings. Intermediate variable calculated in the RADON module. Calculated using an equation on page B-4 in Appendix B of SENES Vol. 3: $EBKDFC = 0.106 \times 10^{-4} \times TVOID \times EXP [-0.261 \times MPCH2O]$ Note: the constants in the above equation are not defined nor described further in the SENES report.
EMANAT	EMANATION COEFFICIENT [-]. Radon emanation coefficient from the upper and lower layers of the unsaturated zone. Input parameter used in the RADON module and sampled from: Triangular (0.02, 0.2, 0.9)
EMRATE(RA)	EMISSION RATE OF RA IN DUST FROM TAILINGS SURFACE [$Bq\ s^{-1}$]. Emission rate of Ra-226 in dust from the tailings surface, as a function of time. Variable calculated in the DUST module. Calculated using equation (1.1) in Appendix B of SENES Vol. 3.

EMRATE(RN)	EMISSION RATE OF RN GAS FROM TAILINGS [Bq s^{-1}]. Emission rate of Rn-222 gas from the tailings unsaturated zone (upper and lower layers) as a function of time. Variable calculated in the RADON module. Calculated using an equation in Appendix B (Section 2) of SENES Vol. 3: EMRATE(RN) = RNFLUX x AEXPT
F	EQUILIBRIUM FACTOR FOR RN+DAUGHTERS [-]. Equilibrium factor for radon daughters to radon. Input parameter used in the DOSE module and described in Appendix F of SENES Vol. 3. Sampled from: Triangular (0.2, 0.4, 0.7)
FACTCO	ACTIVITY COEFFICIENT FOR DIVALENT IONS [-]. Activity coefficient for divalent ions, for use in the unsaturated zone where the ionic stregth is expected to be high. Input parameter used in the TSOURC modules and sampled from: Triangular (0.2, 0.3, 0.5)
FAUT(v)	FRACTION OF ANNUAL INTAKE OF VEGETATION TYPE v [-]. Fraction of annual intake that is local vegetation, with v=1=lichen and v=2=pasture grass. Input parameter used in the DOSE module and sampled from: Triangular (0.05, 0.2, 0.5) (v=1 and 2)
FBI	FEED TO MEAT TRANSFER COEFFICIENT FOR RA-226 [Bq g^{-1} (wet weight) Bq^{-1} a]. Feed-to-meat transfer coefficient specific to Ra-226 and caribou. Input parameter used in the DOSE modules and sampled from: Uniform ($0.3\text{x}10^{-9}$, $2.5\text{x}10^{-9}$)
FDEPED(v)	FRACTION ON EDIBLE PORTION [-]. Fraction of Ra-226 deposited on vegetation type v that is retained on edible portions. Vegetation types v=1,2,3,4 are lichens, grass, above-ground and below-ground vegetables. Input parameters used in the DOSE module and sampled from: Uniform (0.8, 1.0) (v=1,2,3) Uniform (0.05, 0.15) (v=4)
FDEPSF(v)	FRACTION OF DEPOSITION RETAINED [-]. Fraction of total deposition of Ra-226 that is retained on the surface of vegetation type v. Vegetation types v=1,2,3,4 are lichens, grass, above-ground and below-ground vegetables. Input parameters used in the DOSE module and sampled from: Uniform (0.8, 1.0) (v=1) Normal (0.47, 0.3, 0.01, 1.0) (v=2) Uniform (0.1, 0.3) (v=3,4)
FFISH	FRAC. OF ANNUAL INTAKE OF LOCAL FISH [-]. Fraction of annual intake that is local fish. Input parameter used in the DOSE module and sampled from: Constant (1.0)
FIR	FEED INGESTION RATE [g (wet weight) a^{-1}]. Feed ingestion rate for caribou. Input parameter used in the DOSE module and sampled from: Constant ($2.5 \text{ x } 10^{3}$) (as corrected by SENES in a letter dated 13-Feb-1986).
FISH	MODULE. All processes required to calculate Ra-226 concentrations in fish flesh, given concentrations in the lake (see DFD 4 and DDD 4.2). The major output from FISH is CFISH.
FISHDS	DOSE FROM INGESTION OF FISH [Sv a^{-1}]. Annual average effective whole body equivalent commitment due to the ingestion of Ra-226 in fish meat. Variable calculated in DOSE and used as a major output variable. Calculated using equation (2.4) in Appendix F of SENES Vol. 3: FISHDS = DCFING x FISHIN x CFISH x FFISH

FISHIN	AVERAGE ANNUAL INTAKE OF FISH $[g\ a^{-1}]$. Average annual intake of fish. Input parameter used in the DOSE module and sampled from: Uniform ($4x10^{3}$, $12x10^{3}$)
FLOC(v)	FRAC. OF VEG. TYPE v GROWN LOCALLY [-]. Fraction of vegetable type v that is grown locally (v=3 denotes above-ground and v=4 below-ground vegetables). Referred to in Table F.1 of SENES Vol. 3 as FLOCA and FLOCB. Input parameter used in the DOSE modules and sampled from: Triangular (0.1, 0.3, 0.6) (v=3 and v=4) Note that SENES defines FLOCB (=FLOC(4)) to have the same distribution as FLOCA (=FLOC(3)).
FMEAT	FRAC. OF ANNUAL INTAKE OF LOCAL CARIBOU [-]. Fraction of annual caribou meat intake that is local. Input parameter used in the DOSE module and sampled from: Uniform (0.0, 1.0)
FPREC	ANNUAL DAYS WITH SIG. PRECIP. [d]. Annual number of days with more that 0.25 mm of precipitation, excluding winter months. Input parameter used in the DUST module and sampled from: Normal (123, 0.8, 121, 125) (as corrected by SENES in a letter dated 13-Feb-1986).
FRACGP	MASS FRAC. OF GYPSUM IN TAILINGS [-]. Mass fraction of gypsum in tailings. Input parameter used in the TSOURC modules and sampled from: Constant (0.05) (as corrected by SENES on 10/2/86 and in a letter dated 13-Feb-1986).
FRACRA	FRAC. OF RA IN REPRECIP. GYPSUM [-]. Mole fraction of Ra-226 associated with reprecipitated gypsum in the lower (gypsum-enriched) layer of the unsaturated zone, as a function of time. Intermediate variable calculated in the TSOURC modules. Calculated from the iterative finite difference equation (3.24) in Appendix A of SENES Vol. 3. Note that no initial condition is stated for use with equation (3.24), and it is assumed that FRACRA=0 at TIME=0.
FRACV(v)	FRAC. OF RA-226 IN VEG. v AFTER PREPARATION [-]. Fraction of activity in vegetable type v after food preparation (v=3 denotes above-ground and v=4 below-ground vegetables). Referred to in Table F.1 of SENES Vol. 3 as FACTA and FACTB. Input parameter used in the DOSE module and sampled from: Uniform (0.5, 1.0) (v=3 and v=4)
FWAT	FRAC. OF WATER INTAKE THAT IS LOCAL [-]. Fraction of annual water consumption that is local. Input parameter used in the DOSE modules and sampled from: Triangular (0.4, 0.6, 0.9) Note that the input distribution for FWAT is not explicitly stated in Table F.1; rather it is implicitly suggested that FWAT has the same input distribution as OF.
FWIND1	PERCENT OF TIME WITH HIGH WIND SPEEDS [%]. Percentage of the time that the wind speed over the tailings surface exceeds 5.4 m s^{-1} at the mean tailings height (HSRC), for the six summer months. Input parameter used in the DUST module and sampled from: Normal (35, 0.3, 34.1, 35.9) (as corrected by SENES in a letter dated 13-Feb-1986).
FWIND2	FRACTION OF TIME WIND BLOWS TO RECEPTOR [-]. Fraction of the time the wind blows from the tailings source to the receptor (a 'critical' man or vegetation). Input parameter used in the AIRDIS module and sampled from:

	Normal (0.1, 8.0x10^{-4}, 9.73x10^{-2}, 0.103) (as corrected by SENES in a letter dated 13-Feb-1986, and probably containing a 500-year variance).
GYPBLK	INIT. GYPSUM CONTENT IN TAILINGS [mol m^{-3}]. Initial solid phase gypsum content of tailings. This apparently refers to the bulk tailings. Intermediate variable calculated in and used throughout the TSOURC modules. Calculated from equation (2.1) in Appendix A of SENES Vol. 3: GYPBLK = RHOTAL x FRACGP / 136
GYPLFT	MASS OF GYPSUM LEFT IN THE MIXING ZONE [mol m^{-2}]. Mass of gypsum per unit area left in the mixing zone, taking into account the dissolution that begins at TIME = TFINAL. Intermediate variable calculated in the TSOURC modules. Calculated using a modification of equation (7.4) from Appendix A of SENES Vol. 3 (see the DDD for GYPLFT).
HSRC	HEIGHT OF SOURCE [m]. Mean height of the tailings source. Input parameter used in the AIRDIS module and sampled from: Uniform (1.0, 3.0)
ION	CONC. OF IONS FROM GYP. DISS'N IN MIXING ZONE [mol m^{-3}]. Concentration of Ca and SO_4 ions in the aqueous phase of the mixing zone, due to dissolution of gypsum. Intermediate variable calculated in the TSOURC modules. Calculated from equation (7.2) in Appendix A of SENES Vol. 3.
KPRIME	RATIO OF KO_2 TO $[O_2]$ [-]. Ratio of O_2 half-saturation constant to $[O_2]$. Input parameter used in the TSOURC modules and sampled from: Lognormal (5 x 10^{-2}, 1.7, 0.02, 0.25)
KRATE	RATE CONST. OF OXD'N OF PYRITE [a^{-1}]. Rate constant for oxidation of pyrite. Intermediate variable calculated in the TSOURC modules. Calculated from equation (3.13) in Appendix A of SENES Vol. 3: KRATE = SFLUX / (2 x DUNSAT x PYRITE)
KSUBDR	RA AQ-SOLID TRANSFER COEFFICIENT [m^3 g^{-1}]. Radium liquid-solid transfer coefficient (also known as distribution coefficient) in the mixing zone. Input parameter used in the TSOURC modules and sampled from: Lognormal (80.0x10^{-6}, 3.5, 1.9x10^{-6}, 3.4x10^{-3})
KSUBSO	SOLUBILITY EQM. CONST. FOR GYPSUM [mol^2 m^{-6}]. Equilibrium constant for the dissolution of gypsum (solubility product constant). Input parameter used in the TSOURC modules and sampled from: Constant (25.0)
LAKE	MODULE. All processes required to calculate concentrations from lake pathways, given concentrations of relevant ions from the tailings source (see DFD 4 and 4.1, and DDD 4.1.1 to 4.1.3). The major output from LAKE is CLAKE.
LAKEPH	MODULE. All processes required to calculate the pH in the lake, given concentrations of relevant ions from the tailings source (see DFD 4 and 4.3, and DDD 4.3.1 to 4.3.4). The major output from LAKEPH is PHLAKE.
MGYPI	INIT. MASS OF GYPSUM IN MIXING ZONE [mol m^{-2}]. Initial solid phase mass of gypsum, per unit area, in the mixing zone. Intermediate variable calculated in the TSOURC modules. Calculated from equation (7.3) in Appendix A of SENES Vol. 3: MGYPI = GYPBLK x DMIX

MLAKE	MASS OF RA IN LAKE WATER [Bq]. Radium activity in lake water as a function of time. MLAKE and MSED are intermediate variables calculated in the LAKE module, and are calculated from the pair of coupled finite difference equations given in equations (2.5) and (2.6) in Appendix C of SENES Vol. 3. The initial conditions are assumed to be MLAKE = MSED = 0 at TIME = 0. Note that the equations for MLAKE and MSED require that the variable RAMIX be converted from [mol m^{-3}] to [Bq m^{-3}] before it is used in equations (2.5) and (2.6).
MODULE	One or more related processes required to produce one or more calculated variables. The modules referred to here appear as one or more FORTRAN subroutines. Modules used here are based on those noted in Figure 5.1 of SENES Vol. 3 (and which are: SOURCE, LAKE, LAKEPH, RADON, DUST, FISH, AIRDIS, VEGDEP, VEGEAT and DOSE). They were introduced during the construction of this data dictionary and the data dependency diagrams to help with the understanding of the material in Appendices A to F of SENES Vol. 3. Although these modules are not directly reflected as high-level bubbles in the data flow diagrams, they are useful in organizing some groups of processes. The modules referred to in this data dictionary, and their defining Data Flow Diagrams, are as follows: AIRDIS - "Calculate concentrations in air at receptor" (DFD 2.3) DUST - "Calculate Ra emanation in dust at tailings source" (DFD 2.1) DOSE - "Calculate total dose" (DFD 3) FISH - "Calculate Ra concentration in fish" (DFD 4.2) LAKE - "Calculate lake concentration and retention time" (DFD 4.1) LAKEPH - "Calculate lake pH" (DFD 4.3) RADON - "Calculate Rn gas emanation at tailings source" (DFD 2.2) TSOURC - "Calculate tailings source terms" (DFD 1) VEGDEP - "Calculate concentrations in vegetables" (DFD 2.4) VEGEAT - "Calculate concentration in caribou" (DFD 2.5)
MOLTOB	CONVERSION FACTOR MOL TO BQ (RA-226) [Bq mol^{-1}]. Conversion factor used to convert radium concentrations in [mol m^{-3}] to [Bq m^{-3}]. Specifically required for RACON1, RACON2, and RAMIX when these variables are used in modules other than TSOURC. In equation (2.2) of Appendix A, the constant defined there is the reciprocal of MOLTOB. Taking its recprocal, MOLTOB is given by 8.36×10^{12}. MOLTOB is treated as an input parameter sampled from Constant (8.36×10^{12})
MPCH2O	PERCENT MOISTURE CONTENT OF TAILINGS [%]. Percent moisture content by weight of the tailings. Input parameter used in the RADON module and sampled from: Triangular (5, 15, 25)
MSED	MASS OF RA IN LAKE SEDIMENT [Bq]. Radium activity in exchangeable lake sediment as a function of time. MLAKE and MSED are intermediate variables calculated in the LAKE module, and are calculated from the pair of coupled finite difference equations given in equations (2.5) and (2.6) in Appendix C of SENES Vol. 3. The initial conditions are assumed to be MLAKE = MSED = 0 at TIME = 0. Note that the equations for MLAKE and MSED require that the

variable RAMIX be converted from [mol m^{-3}] to [Bq m^{-3}] before it is used in equations (2.5) and (2.6).

NPRCP	NET PRECIPITATION [m a^{-1}]. Net precipitation (accounting for evapotranspiration) on the tailings surface. Input parameter used in the TSOURC modules and sampled from: Normal (0.4, 4.0×10^{-3}, 0.39, 0.41) (as corrected by SENES in a letter dated 13-Feb-1986). ***note: the variance for this parameter should be a 500-year variance. That is, assuming that the annual variation occurs around a long-term mean, the required variance is the variance around a 500-year mean. The value given above is assumed to be the correct value.
O2CONC	O_2 CONC. IN AIR [mol m^{-3}]. Concentration of oxygen in air at the surface of the tailings. Input parameter used in the TSOURC module and sampled from: Constant (9.04)
O2DIFF	O_2 DIFFUSIVITY IN TAILINGS [m^2 a^{-1}]. Diffusion coefficient for the diffusion of oxygen through the unsaturated zone in the tailings. Input parameter used in the TSOURC modules and sampled from: Triangular (30, 70, 130)
OF	FRAC. OF YEAR AT LOCATION [-]. Fraction of the year spent at the receptor site and exposed to inhalation of Ra-226 in dust and Rn gas. Input parameter used in the DOSE module and sampled from: Triangular (0.4, 0.6, 0.9)
PAIRDS	List of input (sampled) parameters required for the calculation of CAIR(RN) and CAIR(RA) and consisting of: XDIST, HSRC, VDISP, FWIND2, UWIND.
PARAMETER	An arbitrary constant or variable so appearing in a mathematical expression that changing it gives various cases of the phenomenon represented (McGraw-Hill Dictionary of Scientific and Technical Terms, 1978). As used here, parameters are input variables to the uranium tailings model, and will be sampled during the stochastic simulation process.
PATMOS	List of input (sampled) parameters required for the calculation of concentrations from atmospheric pathways (the AIRDIS+RADON+DUST+VEGDEP+VEGEAT modules), and consisting of the contents of PAIRDS + PDUST + PRADON + PVGDEP + PVGEAT.
PCHEM	List of input (sampled) parameters required to calculate CHEMICAL CONSTANTS. It consists of DUNSAT, FACTCO, KPRIME, KSUBSO, O2CONC, O2DIFF, NPRCP, PYRITE, RHOTAL, and SULPRD.
PDOSE	List of input (sampled) parameters required for the calculation of dose, and consisting of: DCFING, DCFINH, FFISH, FISHIN, WATIN, FWAT, VEGIN(v), FLOC(v), FRACV(v) (v=3,4), XMEATI, FMEAT, F, OF, XINHIN.
PDUST	List of input (sampled) parameters required for the calculation of Ra emanation in dust at the tailings source (the DUST module) and consisting of: SILT, FPREC, FWIND1, AEXPT, RHOTAL.
PGYP	List of input (sampled) parameters required for the calculation of position of the gypsum-depleted front. It consists of DUNSAT AND NPRCP. The variable DELT is also included, although it is not a sampled parameter. Note: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone.

PHLAKE	PH OF LAKE [-]. pH of the lake as a function of time. Variable calculated in the LAKEPH module and a major output of the tailings submodel. Calculated using the last equation (not reproduced here) in Appendix C of SENES Vol. 3, with a correction noted on 10/2/86.
PHUNST	POREWATER PH IN UNSAT. ZONE [-]. pH of the porewater in the unsaturated zone, as a function of time and depth. Intermediate variable calculated in the TSOURC modules. Note that PHUNST is not required anywhere, and is not a major output variable, although it might be of interest as a minor output variable. It is not included in the specifications and coding done by AECL. Calculated using equation (4.1) in Appendix A of SENES Vol. 3, with a correction noted in discussion with SENES on 10/2/86.
PLAKE	List of input (sampled) parameters used in the calculation of concentrations from lake pathways (the LAKE and LAKEPH modules), and consisting of: ATAIL, XK1, XK3, BETA, NPRCP, VLAKE, AWAT, ACTCO and BF. The variable DELT is also included, although it is not a sampled parameter.
PMIXZ	List of input (sampled) parameters required to calculate chemical concentrations in the mixing zone. It consists of DELT, DMIX, DUNSAT, FACTCO, KSUBSO, NPRCP.
PNONRA	List of input (sampled) parameters required for the calculation of non-Ra chemical changes in the unsaturated zone. It consists of DUNSAT, NPRCP, PCHEM, PGYP, and PMIXZ. Of these, PCHEM, PGYP, and PMIXZ are themselves clusters of parameters. Some parameters appear more than once.
PRADON	List of input (sampled) parameters required for the calculation of Rn gas emanation at the tailings source (the RADON module), and consisting of: ATAIL, RNLAM, EMANAT, RHOTAL, TVOID, MPCH2O.
PRAFTH	List of input (sampled) parameters required for the calculation of Ra-226 concentraions in the liquid phase of the unsaturated zone (Figure 1.3), and consisting of: DUNSAT, NPRCP, RAINTH, RHOTAL, THDISS, THFRAC, THLAM, UO2TON and UPERTH.
PRAMIX	List of input (sampled) parameters required to calculate CONC. OF RA AQ. IN MIXING ZONE. It consists of DISTCO, DMIX, RHOTAL, KSUBDR, NPRCP AND TVOID. The variable DELT is also included, although it is not a sampled parameter.
PRAUP	List of input (sampled) parameters required to calculate Ra concentrations, particularily RACON1, in the upper unsaturated zone. It consists of CONCRA, RAINGP, RALAM, RHOTAL, PRAFTH. PRAFTH is itself a list of parameters that partially overlaps this list.
PSOURC	List of input (sampled) parameters required by the TSOURC modules, and consisting of: RAINGP, RAINTH, THFRAC, THLAM, RALAM, RNLAM, RHOTAL, CONCRA, DUNSAT, NPRCP, FRACGP, KSUBSO, FACTCO, PYRITE, O2CONC, O2DIFF, KPRIME, SULPRD, DISTCO, UO2TON, UPERTH, THDISS, DMIX, TVOID, KSUBDR. The variable DELT is also included, although it is not a sampled parameter.
PVGDEP	List of input (sampled) parameters required for the calculation of CVEG(v) (in the VEGDEP module), and consisting of DEPVEL, FDEPSF(v), FDEPED(v), VEGLAM(v), DUREXP(v), YDEN(v); v=1,2,3,4.
PVGEAT	List of input (sampled) parameters required for the calculation of CMEAT (in the VEGEAT module), and consisting of FIR, FBI, FAUT(v); v=1,2.

PYRITE	INIT. CONC. OF PYRITE IN TAILINGS [mol m^{-3}]. Initial concentration of pyrite in the tailings. Input (sampled) parameter used in the TSOURC modules and sampled from: Constant (730).
QLAKE	FLOW IN & OUT OF LAKE [m^3 a^{-1}]. Annual flow in and out of the lake. Intermediate variable calculated in the LAKE module. Calculated from an expression near equation (3.2) in Appendix C of SENES Vol. 3: QLAKE = NPRCP x AWAT
R	[mol m^{-3}]. Intermediate variable defined in the equation and DFD for TFINAL in the TSOURC module. Calculated using: R = CSINIT / 2
RAAQ	CONC. OF RA AQ IN POREWATER OF UNSAT. ZONE [mol m^{-3}]. Concentration of Ra-226 in the aqueous phase of the porewater of the unsaturated zone. Calculated using equation (3.2) in Appendix A of SENES Vol. 3: RAAQ = DISTCO x FRACRA x CONCCA
RACON1	RA CONC. IN UPPER UNSAT. ZONE [mol m^{-3}]. Concentration of Ra-226 in the solid phase in the upper (gypsum-depleted) layer of the unsaturated zone. Calculated in the TSOURC modules, and used in the DUST and RADON modules. RACON1 is calculated from equation (6.1) of Appendix A of SENES Vol. 3. Note 1: when RACON1 is required in the DUST and RADON modules, it must be used in units of [Bq m^{-3}], as noted in a conversation with SENES on 10/2/86. To maintain a consistent set of variable definitions, MOLTOB x RACON1 should be passed to these modules, where MOLTOB performs the unit conversion. Note 2: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone.
RACON2	RA CONC. IN LOWER UNSAT. ZONE [mol m^{-3}]. Concentration of Ra-226 in the solid phase in the lower (gypsum-enriched) layer of the unsaturated zone. Calculated in the TSOURC modules, and used in the RADON module. RACON2 is calculated from equation (6.2) of Appendix A of SENES Vol. 3. Note 1: when RACON2 is required in the DUST and RADON modules, it must be used in units of [Bq m^{-3}], as noted in a conversation with SENES on 10/2/86. To maintain a consistent set of variable definitions, MOLTOB x RACON2 should be passed to these modules, where MOLTOB performs the unit conversion. Note 2: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone.
RADIUM	TOTAL INIT. CONC. OF RA IN SOLIDS [mol m^{-3}]. Total initial solid phase concentration of Ra-226 in the tailings. Intermediate variable calculated in and used throughout the TSOURC modules. RADIUM is calculated using equation (2.2) from Appendix A of SENES Vol. 3.
RADON	MODULE. All process required to calculate the Rn gas emanation from the tailings source (see DFD 2, 2.2 and DDD 2.2.1 to 2.2.3). The major output from RADON is EMRATE(RN).
RAFRNI	INIT. MF OF RA IN GYPSUM [-]. Initial mole fraction (more accurately, fraction) of RA-226 in gypsum. Intermediate variable calculated in the TSOURC modules. Calculated from an expression near equation (2.2) in Appendix A of SENES Vol. 3: RAFRNI = CRAGIN / GYPBLK

RAFTH	CONC. OF RA AQ FROM TH DECAY IN UNSAT. ZONE [mol m^{-3}]. Concentration of Ra-226 in the aqueous phase of the unsaturated zone, due to the decay of Th-230. Intermediate variable calculated in the TSOURC modules. Calculated using equation (5.5) in Appendix A of SENES Vol. 3. Note that RAFTH appears in several different formulations in the SENES Report.
RAINGP	FRAC. OF TOTAL RA IN GYPSUM [-]. Fraction of the total Ra-226 this is associated with gypsum. Input (sampled) parameter used in the TSOURC modules and sampled from: Triangular (0.05, 0.2, 0.6)
RAINTH	FRAC. OF INGROWN RA THAT IS INSOL. [-]. Mole fraction of Ra-226 ingrowth to solid Th-230 matrix that is insoluble. Input (sampled) parameter used in the TSOURC modules and sampled from: Triangular (0.85, 0.90, 0.95)
RALAM	RA-226 DECAY CONSTANT [a^{-1}]. Radioactive decay constant for Ra-226. Input parameter used in the TSOURC module, and sampled from: Constant (4.33×10^{-4}) (as defined on page A-4 of SENES Vol. 3).
RAMIX	CONC. OF RA AQ IN MIXING ZONE [mol m^{-3}]. Concentration of Ra-226 in the aqueous phase of the mixing zone. Note 1: it is assumed that RAMIX is also equivalent to the variable: "concentration of radium in the contaminated surface runoff from the tailings area" referred to in Appendix C on page C-2. Variable calculated in the TSOURC modules, and used in the LAKE module. Calculated from the finite difference equation (7.1) in Appendix A of SENES Vol. 3, assuming that RAMIX=0 at TIME=0. Note 2: equation (7.1) was corrected following discussion with SENES; see the DDD for RAMIX. Note 3: when RAMIX is required in the LAKE module, it must be used in units of [Bq m^{-3}], as noted in a conversation with SENES on 10/2/86. To maintain a consistent set of variable definitions, MOLTOB x RAMIX should be passed to these modules, where MOLTOB performs the unit conversion.
RARET	RA RETARDATION CONST. IN MIXING ZONE [-]. Radium retardation constant, describing the relative rate of movement of radium to that of water. Intermediate variable defined in the DDD for RAMIX using: RARET = 1 + KSUBDR x RHOTAL / TVOID
RASIMM	INIT. CONC. OF IMMOBILE RA IN TAILINGS [mol m^{-3}]. Initial concentration of Ra associated with host tailings material assumed not to be leachable. Intermediate variable calculated and used in the TSOURC modules. Calculated (from an expression near equation (2.2) in Appendix A of SENES Vol.3): RASIMM = (1-RAINGP)*RADIUM
RATH23	CONC. OF RA-226 DUE TO INGROWTH FROM TH-230 [mol m^{-3}]. Concentration of Ra-226 in the solid phase due to ingrowth from the Th matrix. Intermediate variable calculated and used in the TSOURC modules. Calculated from equation (5.6) in Appendix A of SENES Vol. 3.
REPGYP	REPREC. GYPSUM MASS IN LOWER UNSAT. ZONE [mol m^{-2}]. Reprecipitated gypsum content per unit area in the bottom (gypsum-enriched) layer of the unsaturated zone, as a function of time and depth. Intermediate variable calculated in the TSOURC modules. Calculated using equation (3.22) in Appendix A of SENES Vol. 3.

REPRA	MASS OF RA IN REPREC. GYPSUM [mol m^{-2}]. Radium content of reprecipitated gypsum per unit area in the bottom (gypsum-enriched) layer of the unsaturated zone, as a function of time and depth. Intermediate variable introduced in analysis to simplify data flow diagrams. REPRA = FRACRA*REPGYP
RHOTAL	TAILINGS BULK DENSITY [g m^{-3}]. Bulk density of the tailings. Input parameter used in the TSOURC modules and sampled from: Normal (1.4×10^{6}, 0.1×10^{6}, 1.1×10^{6}, 1.7×10^{6})
RNDOS	DOSE FROM INHALATION OF RN [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to the inhalation of Ra-222 gas and its radioactive daughters. Variable calculated in the DOSE module and a major output variable. Calculated using equation (3.2) in Appendix F of SENES Vol. 3: RNDOS = 1.09 x 10^{-4} x F x OF x CAIR(RN) where the constant is related to the conversion of working level months to Sieverts, etc. (see Appendix F of SENES Vol. 3).
RNFLUX	RN FLUX FROM THE ENTIRE UNSATURATED ZONE [Bq m^{-2} s^{-1}]. Ra-222 flux from the upper (gypsum-depleted) and lower (gypsum-enriched) layers of the unsaturated zone, as a function of time. Intermediate variable calculated in the RADON module. Calculated using equation (2.4) in Appendix B of SENES Vol. 3. Note: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone.
RNLAM	RN-222 DECAY CONSTANT [s^{-1}]. Radioactive decay constant for Rn-222. Note that the units of RNLAM are different from the units of RALAM and THLAM; however the appropriate equations have been checked for consistency. Input parameter used in the RADON module and sampled from: Constant (2.1 x 10^{-6})
S	[mol m^{-3}]. Intermediate variable defined in the equation and DFD for DEPTH1 in the TSOURC module. Calculated using: S = CSINIT x DEPTH1 / (2 x DUNSAT)
SFLUX	SULPHATE FLUX FROM OXD'N OF PYRITE [mol m^{-2} a^{-1}]. Net sulphate flux out of the bottom of the unsaturated zone (and into the mixing zone) due to the oxidation of pyrite and the downward movement of water. Intermediate variable calculated in the TSOURC modules, using equation (3.12) in Appendix A of SENES Vol. 3.
SILT	SILT CONTENT OF TAILINGS SURFACE [%]. Percentage of particles in the tailings smaller than 75 x 10^{-6} m. Input parameter used in the DUST module and sampled from: Constant (30)
SO4MIX	CONC. OF SO4 AQ IN THE MIXING ZONE [mol m^{-3}]. Concentration of SO_4 in the aqueous phase of the mixing zone. Variable calculated in the TSOURC modules and used in the LAKEPH module. Calculated using equation (7.1) and others nearby in Appendix A of SENES Vol. 3: SO4MIX = CONCS8 (TIME <= TFINAL) SO4MIX = CONCS8 + ION (TIME > TFINAL and GYPLFT > 0) SO4MIX = CONCS8 (TIME > TFINAL and GYPLFT <= 0) Note: all values of CONCS8 are required at a depth equal to DUNSAT.

SULAKE	CONC OF SO4 IN LAKE [mol m^{-3}]. Concentration of sulphate in the lake as a function of time. Intermediate variable calculated in the LAKEPH module. Calculated from a finite difference equation near equation (3.2) in Appendix C of SENES VOL. 3, with the unstated assumption that SULAKE = 0 at TIME = 0. Note: to keep consistency with other units, SULIN must be used in [mol m^{-3}] when this finite difference equation is used.
SULIN	SO4 INFLOW TO LAKE [mol m^{-3}]. Concentration of sulphate in the aqueous phase entering the lake from the mixing zone and from background levels. Intermediate variable calculated in the LAKEPH module. Calculated using an equation before equation (3.1) in Appendix C of SENES Vol. 3: $SULIN = 2.5 \times 10^{-2} \times SO4MIX + 10^{-1.5}$ Note 1: the first constant in the above equation is a dilution factor, and the second is the "background" level of SO_4 concentrations; see SENES Vol. 3, p C-5. Note 2: to keep consistency with other variables, SULIN must be used in [mol m^{-3}] when used to calculate SULAKE. This requires the conversion of background sulphate concentrations from [mol l^{-1}] to [mol m^{-3}], which has been done in the above equation.
SULPRD	SULPHATE PRODUCTION RATE [mol g^{-1} a^{-1}]. Sulphate production rate. Input parameter described in Appendix A of SENES Vol. 3. Sampled from: Lognormal $(1.4\times10^{-6}, 1.5, 0.4\times10^{-6}, 4.2\times10^{-6})$
S1	RN ACTIVITY EMANATING FROM THE UPPER UNSAT. ZONE [Bq m^{-3} s^{-1}]. Proportional to the rate of emission of Rn-222 per unit area from the upper (gypsum-depleted) layer of the unsaturated zone. Intermediate variable calculated in the RADON module. Following conversations with SENES on 10/2/86, the following equation should be used (adapted from an equation near equation (2.2) in Appendix B of SENES Vol. 3): $S1 = MOLTOB \times RACON1 \times RNLAM \times EMANAT$ where RACON1 has been converted from [mol m^{-3}] to [Bq m^{-3}] using the conversion factor MOLTOB. Note: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone.
S2	RN ACTIVITY EMANATING FROM THE LOWER UNSAT. ZONE [Bq m^{-3} s^{-1}]. Proportional to the rate of emission of Rn-222 per unit area from the lower (gypsum-enriched) layer of the unsaturated zone. Intermediate variable calculated in the RADON module. Following conversations with SENES on 10/2/86, the following equation should be used (adapted from and equation near equation (2.2) in Appendix B of SENES Vol. 3): $S2 = MOLTOB \times RACON2 \times RNLAM \times EMANAT$ where RACON2 has been converted from [mol m^{-3}] to [Bq m^{-3}] using the conversion factor MOLTOB.
TDOSE	TOTAL ANNUAL DOSE [Sv a^{-1}]. Total annual average effective whole body dose equivalent commitment due to ingestion and inhalation of Ra-226, and inhalation of Rn-222. Variable calculated in the DOSE module and a primary output variable. Calculated using the following equation (not given in SENES Vol. 3): $TDOSE = TVEGDS + XMEATD + WATDOS + FISHDS + XINDOS + RNDOS$

TFINAL	TIME TO DEPLETE GYPSUM IN UNSAT. ZONE [a]. Time at which the entire unsaturated zone is depleted of gypsum (and when DEPTH1 = DUNSAT). Intermediate variable calculated in and used throughout the TSOURC modules. Calculated using iterative equation (3.21) in Appendix A of SENES Vol. 3.
THDISS	RATE OF AMORPHOUS TH DISSOLUTION [a^{-1}]. Rate constant for the dissolution of amorphous thorium oxide. Also called an "arbitrary" rate constant in Appendix A (Section 5.1.1) of SENES Vol. 3. Sampled from: Uniform (0.01, 0.03)
THFRAC	FRAC. OF IMMOBILE TH IN TAILINGS [-]. Fraction of total TH ores which is assumed to be immobile (ie. fraction of Th assumed to be in a crystalline rather than an amorphous form). Input parameter described in Appendix A of SENES Vol. 3 and sampled from: Triangular (0.2, 0.3, 0.4)
THINIT	INIT. TOTAL TH IN SOLID PHASE [mol m^{-3}]. Initial concentration of total Th in the solid phase of the unsaturated zone (cf. THSOLD). Intermediate variable calculated in TSOURC. Calculated using equation (5.2) in Appendix A of SENES Vol. 3: THINIT = CTH230 + CTH232
THINTH	MF OF TH-230 TO TH-232 [-]. Mole fraction (more accurately, fraction) of Th-230 to Th-232 in the unsaturated zone. Intermediate variable calculated in TSOURC. Calculated using equation (2.6) in Appendix A of SENES Vol. 3: THINTH = CTH230 / CTH232
THLAM	TH-230 DECAY CONSTANT [a^{-1}]. Radioactive decay constant for Th-230. Input parameter used in the TSOURC modules and sampled from: Constant (8.66×10^{-6}) (as defined on page A-4 of SENES Vol. 3).
THSOLD	TOTAL TH IN SOLID PHASE [mol m^{-3}]. Concentration of total TH in the solid phase of the unsaturated zone. Intermediate variable calculated in the TSOURC modules. Calculated using equation (5.2) in Appendix A of SENES Vol. 3.
TIME	TIME [a]. Independent variable, implicitly included in the calculation of most variables, and explicitly appearing in the calculation of some variables. Used in many of the modules. Table 5.1 of SENES Vol. 3 suggests that TIME ranges from 0 to 500 in steps (DELT) of 10 a. Major output variables in the tailings submodel are output every 50 a.
TSOURC	MODULES. All processes required to calculate the tailings source terms which are used in other modules. Usually referred to in the plural, since the calculation processes in TSOURC are very complex and long. The major variables produced by TSOURC include RACON1, RACON2, DEPTH1, DEPTH2, RAMIX, CAMIX, and SO4MIX.
TVEGDS	DOSE FROM INGESTION OF VEGETABLES [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to ingestion of Ra-226 in vegetables (above-ground and below-ground types only). Variable calculated in DOSE and a major output variable. Calculated using part of equation (2.1) in Appendix F of SENES Vol. 3: TVEGDS = VEGDOS(3) + VEGDOS(4)
TVOID	PORE VOLUME FRACTION [m^3 m^{-3}]=[-]. Fraction of void space in the mixing zone. Input parameter described in Appendix A of SENES Vol. 3 and sampled from: Triangular (0.35, 0.40, 0.45)

Note: TVOID cancels when used in the equations for EBKDFC and DIFLEN, and so is a spurious parameter when used with these parameters. TVOID is required, however, in the calculation of RAMIX.

UNSAT. ZONE: Abbreviation used to designate the unsaturated portion of the tailings pile. As shown in Figure A.1 of Appendix A in SENES Volume 3, the unsaturated zone consists of an upper (gypsum-depleted) layer of depth = DEPTH1 and radium concentration = RACON1, and a lower (gypsum-enriched) layer of depth = DEPTH2 and radium concentration = RACON2. Precipitation falling on the surface of the tailings passes through the unsaturated zone to the saturated zone below. (The saturated zone consists of a mixing layer and other unspecified layers.) Note: the term gypsum-depleted is somewhat confusing; gypsum-free would be more consistent with the model of the unsaturated zone.

UO2TON INIT. MF OF UO_2 IN ORE [kg(UO_2) tonne^{-1}(ore)]. Initial (mass) fraction of UO_2 per tonne of ore. Input parameter described in Appendix A of SENES Vol. 3 and sampled from:
Constant (1.2)
(but see apparent discrepancy in Table A.1 for the nominal value of UO2TON)

UPERTH MASS RATIO OF UO_2 TO THO_2 [-]. Mass ratio of the UO_2 to THO_2 present in ore (which is assumed to be the residual tailings material). Input parameter described in Appendix A of SENES Vol. 3 and sampled from:
Constant (3.0)

UWIND MEAN WIND SPEED, SOURCE TO RECEPTOR [m s^{-1}]. Mean annual wind speed. Input parameter used in the AIRDIS module and sampled from:
Normal (5.3, 2.2×10^{-2}, 5.2, 5.4) (as corrected by SENES in a letter dated 13-Feb-1986).

VARIABLE A quantity (or a symbol representing a quantity) that may assume any one of a set of values (Webster's Ninth New Collegiate Dictionary, 1983). As used here, variable includes input (sampled) parameters (such as NPRCP), calculated variables (such as RACON1), output variables (such as TDOSE), and input independent variables (such as TIME and DELT).

VDISP VERTICAL DISPERSION COEFFICIENT [m]. Vertical dispersion coefficient used in the sector-averaged Gaussian plume model for the dispersion of contaminants in the atmosphere. Input parameter used in the AIRDIS module and sampled from:
Lognormal (60.0, 1.08, 47.6, 75.7)

VEGDEP MODULE. All processes required to calculate the concentrations of Ra-226 in edible vegatables due to foliar retention of Ra-226 deposited from air (see DFD 2 and DDD 2.4). The major output from VEGDEP is CVEG(v); v=1,2,3,4.

VEGDOS(v) DOSE FROM INGESTION OF VEGETATION TYPE v [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to ingestion of Ra-226 in above-ground (v=3) and below-ground (v=4) vegetables. Intermediate variables calculated in DOSE. Calculated using part of equation (2.1) from Appendix F of SENES Vol. 3:

$$\mathrm{VEGDOS(v) = DCFING \times VEGIN(v) \times CVEG(v) \times FRACV(v) \times FLOC(v)}$$

VEGEAT MODULE. All processes required to calculate the concentration of Ra-226 in caribou meat due to the ingestion of contaminated lichen and grass (see DFD 2 and DDD 2.5). The major output from VEGEAT is CMEAT.

VEGIN(v)	AVERAGE ANNUAL INTAKE OF VEG.v [g(wet wt) a^{-1}]. Average annual intake of vegetable type v=3 (above- ground) and v=4 (below-ground). Referred to as VEGINA and VEGINB in SENES Vol. 3, table F.1. Input parameter used in the DOSE module and sampled from: Constant (1.57×10^5) (v=3) Constant (7.0×10^4) (v=4)
VEGLAM(v)	WEATHERING LOSS DECAY RATE [s^{-1}]. 'Decay' constant accounting for weathering loss in vegetable types v=1,2,3,4. Input parameters used in the VEGDEP module and sampled from: Uniform (2×10^{-9}, 4×10^{-9}) (v=1) Uniform (4×10^{-7}, 6×10^{-7}) (v=2,3,4)
VLAKE	LAKE VOLUME [m^3]. Volume of the lake (which is assumed to be time-independent). Input parameter used in the LAKE module, and sampled from: Constant (1.6×10^7) Note: VLAKE is not in the input file of Table 5.1 of SENES Vol. 3, nor in the updated input file they provided in a letter dated 13-Feb.-1986. It is listed as an input parameter in Table C.1. Note also that if VLAKE = ALAKE x DLAKE (lake area x lake depth), then VLAKE will have a constant value equal to that given above, since ALAKE is given a value of 1.6×10^6 and DLAKE a value of 10.0 in Table C.1 of SENES Vol. 3 and in their letter dated 13-Feb.-1986.
WATDOS	DOSE FROM INGESTION OF WATER [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to the ingestion of Ra-226 in drinking water from a lake. Variable calculated in the DOSE module and a major output variable. Calculated using equation (2.3) in Appendix F of SENES Vol. 3: WATDOS = DCFING x WATIN x CLAKE x FWAT
WATIN	AVERAGE ANNUAL INTAKE OF WATER [m^3 a^{-1}]. Average annual intake of water. Input parameter used in the DOSE modules and sampled from: Constant (0.6)
XDIST	DISTANCE FROM SOURCE TO RECEPTOR [m]. Downwind distance from the tailings source to the receptor. Input parameter used in the AIRDIS module and sampled from: Constant (3600)
XINDOS	DOSE FROM INHALATION OF DUST [Sv a^{-1}]. Annual average effective whole body dose equivalent commitment due to the inhalation of Ra-226 in dust. Variable calculated in the DOSE module, and a major output variable. Calculated using equation (3.1) in Appendix F of SENES Vol. 3: XINDOS = DCFINH x XINHIN x CAIR(RA) x OF
XINHIN	AVERAGE ANNUAL INHALATION RATE [m^3 a^{-1}). Annual average effective inhalation rate for the 'critical' man. Input parameter used in the DOSE module and sampled from: Constant (8400)
XK1	RATE OF RA LOSS TO SEDIMENT [a^{-1}]. Rate of radium removal from the lake water column to lake sediment. Input parameter used in the LAKE module and sampled from: Lognormal (0.43, 1.85, 0.126, 1.47) (as corrected by SENES in a letter dated 13-Feb-1986).

XK2	RATE OF RA GAIN FROM SEDIMENT $[a^{-1}]$. Rate of radium release to the lake water column from the lake sediment. Intermediate variable calculated in the LAKE module. Calculated from an expression not in Vol. 3 of the SENES report, according to: XK2 = BETA x XK1
XK3	RATE OF RA BURIAL IN DEEP SED. $[a^{-1}]$. Rate of radium removal (burial) to deep lake sediment. Input parameter used in the DOSE module and sampled from: Lognormal (0.022, 1.2, 0.0153, 0.0317) (as corrected by SENES in a letter dated 13-Feb-1986).
XLAML	RETENTION TIME IN LAKE [a]. Retention time (or travel time) in the reference lake. Variable calculated in the LAKE module and used in the LAKE and LAKEPH modules. Calculated from equation (3.2) in Appendix C of SENES Vol. 3: XLAML = VLAKE / QLAKE
XMEATD	DOSE FROM INGESTION OF CARIBOU $[Sv\ a^{-1}]$. Annual average effective whole body dose equivalent commitment due to the ingestion of Ra-226 in caribou meat. Variable calculated in the DOSE module and a major output variable. Calculated using equation (2.2) in Appendix F of SENES Vol. 3: XMEATD = DCFING x XMEATI x CMEAT x FMEAT
XMEATI	AVERAGE ANNUAL INTAKE OF CARIBOU MEAT $[g\ a^{-1}]$. Annual intake of meat for an Ontario male, age 20 to 39, assumed to be all caribou. Input parameter used in the DOSE module and sampled from: Constant (7.9×10^{4})
YDEN(v)	YIELD DENSITY $[g(\text{wet weight})\ m^{-2}]$. Mass of vegetation type v (v=1,2,3,4) produced per unit area of ground. Input parameter used in the VEGDEP module, and sampled from: Uniform (400, 600) (v=1) Uniform (500, 1000) (v=2) Uniform (1500, 2500) (v=3,4)
Z	POSITION [m]. Position within the unsaturated zone of the tailings. This is an independent variable appearing in some of the equations (such as for CONCS8), but that is not required since all references to Z are for a fixed value: Z=DUNSAT. The only known exception involves the variable PHUNST, which is not a major output variable.

APPENDIX D. RECORDS FROM THE QUALITY ASSURANCE WALKTHROUGHS FOR THE SPECIFICATIONS

A walkthrough is defined by Page-Jones[1] to be "a peer group review of a product for the purpose of finding any deficiencies in that product". The object of a walkthrough is to find problems or potential problems, and not to find solutions. The attendees are challenged to find flaws.

Three of the attendees have special roles. The presenter presents the product being reviewed and described its function and how it works. The coordinator keeps the meeting focussed on the purpose of the walkthroughs. Finally the recorder writes down each question and critical comment that arise during the meeting. The rest of the participants concentrate on the product, identifying missing elements, unworkable segments, special cases that may not be treated properly, inconsistent logic, and so forth.

The specifications from the uranium tailings model were subjected to a series of formal walkthroughs. The participants were normally provided with the relevant information required to assess the specifications, such as the appropriate appendices from the SENES report[2] , a copy of the data dictionary, and the existing (preliminary) specifications.

At the end of the meeting, the participants were asked to either accept or reject the product. If a product was rejected, follow-up walkthroughs were held until the product was judged to be acceptable. A product can be accepted if, and only if, all participants concur.

The following pages contain a record of the walkthroughs held for the specifications prepared for the uranium tailings model. Each walkthrough consists of a WALKTHROUGH SUMMARY sheet and the supplementary COMMENTS SHEET that contains the commments and critisisms of the participants. The records provided include:

			Page
1.	Review #1 (Pass 1)	February 11, 1986	D-3
2.	Review #1 (Pass 2)	February 17, 1986	D-5
3.	Review #2	February 18, 1986	D-9
4.	Review #3 (Pass 1)	February 20, 1986	D-11
5.	Review #3 (Pass 2)	February 21, 1986	D-14
6.	Review #4 (Pass 1)	February 19, 1986	D-17
7.	Review #4 (Pass 2)	Feb. 25 - Mar. 3, 1986	D-21

Review #4, which dealt with the tailings model source term module, did not reach the final "acceptance" stage of the walkthroughs. Two factors contributed to this incompletion of the walkthroughs: the overall com-

---

1/ Page-Jones, M., 1980. The Practical Guide to Structured Systems Design; YOURDON Inc., New York, P. 294.

2/ SENES, 1985. Probabilistic model development for the assessment of the long-term effects of uranium mill tailings in Canada - Phase II; Volume 3 - Probabilistic analysis user's manual and documentation for the Uranium Tailings Assessment Program (UTAP).

plexity of the source modules, and the shortage of time available. At the end of the second pass, it was decided that most of the major issues were adequately covered, and that coding could begin. It was expected that subsequent passes could take place later on in the project, but this was not possible because of time constraints. Nevertheless, we conclude that walkthroughs are very important tools in the preparation of quality assured products. Many of the problems subsequently encountered with coding and testing of the source modules may have been avoided, if these walkthroughs had been taken to completion.

WALKTHROUGH SUMMARY

REVIEW NUMBER	1 (Pass #1)	STARTING TIME	1:30
DATE	86-Feb-11	ENDING TIME	2:30

PRODUCT IDENTIFICATION NUTP Tailings Model

PRODUCED BY B. Goodwin

BRIEF DESCRIPTION dfds and ddds* for VEGEAT, VEGDEP, and DOSE sections

MATERIALS USED IN THE REVIEW: *(check here ______ if supplementary list)*

IDENTIFICATION	DESCRIPTION
Appendix E, section 1	terrestrial models
Appendix F, sections 1-3	ingestion dose, inhalation dose
dfd	NUTP tailings model
dfd & ddd	VEGEAT
dfd & ddd	VEGDEP
dfd & 6ddds	DOSE

PARTICIPANTS:

	NAME	SIGNATURE
LEADER	T. Andres	
RECORDER	K. Hoffman	Karen Hoffman
PRESENTER	B. Goodwin	
4.	M. Sheppard	Marsha I. Sheppard
5.	R. Zach (written comments via M. Sheppard)	
6.		
7.		

APPRAISAL OF THE WORK UNIT:

ACCEPTED *(no further review)*	NOT ACCEPTED *(new review required)*
______ as is	X major revisions
______ with minor revisions	______ rebuild
	______ review not completed

SUPPLEMENTARY MATERIALS PRODUCED:		DESCRIPTION AND/OR IDENTIFICATION
X	COMMENT SHEETS	attached
X	RELATED ISSUES LIST	retained by M. Sheppard for discussion with R. Holmes
	OTHER	

* dfd is data flow diagram
ddd is data dependency diagram

Walkthrough of the vegetation and Dose Modules [Review #1 (Pass 1)]

(Comments recorded by K.J. Hoffman February 11, 1986)

KH - don't need separate diagram for VEGDEP

KH - need data dictionary

KH - name bubble on VEGDEP dfd - all processes need names

MS - model given to Bruce has many serious inconsistencies

KH - terminology "module" - subroutine or function

BG - "module" to him means above or group of above

MS - code names add complication

KH - need minispec, or to reference ddd as replacing minispec

MS - would help to know source term

TA - thinks we should have looked at earlier processes first

MS - to be able to check inconsistencies

TA, KH - DDD's are presented very well - definitions + equations + unit check all at once

BG - VEGEAT has same already identified problems as VEGDEP

TA - in what sense does time dependence come in - not clear what is and isn't

KH - need better names for data groups - these should be in dictionary

TA - on top diagram, CVEG 1&2 go to VEGEAT, 3&4 to DOSE

KH - XINDOS & RNDOS are reversed on DOSE diagram

KH - can combine calc. VEGDS(v) & calc. TVEGDS into one process

KH, MS - don't need FISHDS, etc to come out of calc. TDOSE if they aren't changed there

KH, TA - major flow should be up not down

TA - top diagram has too many bubbles

TA - need quality circles, ellipses, boxes for final version

WALKTHROUGH SUMMARY

REVIEW NUMBER 1 (Pass #2) STARTING TIME 12:00p.m. - 12:15 p.m.

DATE 86 - Feb - 17 ENDING TIME 13:05p.m. - 13:35 p.m.

PRODUCT IDENTIFICATION NUTP Tailings Model

PRODUCED BY B.W. Goodwin

BRIEF DESCRIPTION dfds and ddds for VEGEAT, VEGDEP and DOSE sections

MATERIALS USED IN THE REVIEW: *(check here ______ if supplementary list)*

IDENTIFICATION	DESCRIPTION
Appendix E, Section 1	terrestrial models
Appendix F, Sections 1-3	ingestion dose, inhalation dose
dfd	NUTP tailings model overview
dfd & ddds	modified from review #1

PARTICIPANTS:

	NAME	SIGNATURE
LEADER	T.H. Andres	
RECORDER	T.H. Andres	
PRESENTER	B.W. Goodwin	BWGood
4.	M.I. Sheppard	{ oral comments recorded M.I. Sheppard
5.	R.R. Zach	by BWG
6.	K.J. Hoffman	Ken Hoffman
7.		

APPRAISAL OF THE WORK UNIT:

ACCEPTED *(no further review)*

______ as is

x with minor revisions

NOT ACCEPTED *(new review required)*

______ major revisions

______ rebuild

______ review not completed

SUPPLEMENTARY MATERIALS PRODUCED: DESCRIPTION AND/OR IDENTIFICATION

X COMMENT SHEETS attached

______ RELATED ISSUES LIST

______ OTHER

Walkthrough of the Vegetation and Dose Modules [Review #1 (Pass 2]

(Comments recorded by B.W. Goodwin, February 17, 1986)

Comments from T.H. Andres:

1. "Notes for data flow diagrams"
- jargon terms DFD's, DDD, mini-spec (what is it?)
- "for example the DFD 'calculate concentration...'"; use "process bubble" instead of DFD here
- time-dependence: still ambiguous which data flows are time-dependent.
- use "calculated variables", not "calculated parameters"
- non-standard use of "module" to refer to a cluster of subroutines; what benefit is this non-standard use?
- these notes are good to have to explain the data flow diagrams.

2. Figure 0: Context diagram
- data dictionary entries provide redundant information about where data flows. Not necessary, and hard to keep up. Exception: if this info refers only to SENES code, this fact should be clearly indicated.
- PDOSE missing from data dictionary.
- DEPTH1: DD def'n: move over "(TIME >= TFINAL)".
- DEPTH2: last equation line is redundant and can be deleted.
- PHLAKE: not defined in DD
- PSOURC: has extra "not" typo in DD.
- RACON1, RACON2: unit conversion should appear in formula, not in DD.
- RAMIX: def'n uses DDD jargon.
- SO4MIX: typo involving GYPLFT in DD.
- alphabetical order in DD: SFLUX, SO4MIX, S1, S2, SILT

3. Figure 2:
- in all diagrams, put number in top right hand corner for easier scanning.
- PAIRDS: missing from DD and from PATMOS def'n.
- in all process bubbles, make specific names match DD names; e.g. EMRATE(Ra) is "Ra emanation in dust at tailings source" in 2.1, but "EMISSION RATE OF RA IN DUST FROM TAILINGS SURFACE" in DD.

4. Figure 2.4:
- in all the "minispec" pages, the number should appear in the top right hand corner and in the oval bubble for the final data flow, the definition at the top should include the long variable name from the DD, and the definition should match the DD def'n.
- CAIR(RA): def'n should include the words "average annual"; the oval bubble should include "(Ra)" to match the DD long name.

- FDEPSF(v): no DD def'n.
- FDEPED(v): no DD def'n.
- YDEN: def'n should mention "wet weight".
- How does Note 2 relate to this minispec?

5. Figure 2.5:
- CMEAT: def'n should mention "average annual" and "wet weight".
- FIR: missing from DD.
- FBI: missing from DD.
- FAUT(v): missing from DD.

6. Figure 3.0:
- arrows for outgoing data flows should approach edge of page for easier detection.
- names F, OF are too short to be meaningful.
- OF: def'n missing in DD.
- VEGIN: missing from DD.
- FLOC(v), FRACV(v): syntax "v=3=above-ground" is poor; cf "v=3 means above-ground".

7. Figure 3.1: No comments.

8. Figure 3.2:
- FISHDS: def'n should use "average".
- CFISH: def'n should use "average annual".

9. Figure 3.3:
- WATDOS: def'n should use "average".

10. Figure 3.4:
- TVEGDS: def'n should use "average" and "effective"
- CVEG(v): suggest def'n should use "CONC. OF CONTAMINANT (Ra-226)...", and should mention "wet weight".

11. Figure 3.5:
- XMEATI: def'n should mention "Ontario male", age, etc.
- CMEAT: long name does not include "(Ra-226)", as shown on diagram. Def'n should include "average annual"
- XMEATD: def'n should use "average".

12. Figure 3.6:
- RNDOS: def'n should use "average"
- 1.09×10^{-4}: should be made parameter, or product of parameters as discussed in SENES text.

13. Figure 3.7:
- CAIR(Ra): name does not include "-226" as shown; def'n should use "average annual".

Comments from M.I. Sheppard

Note: M.I. Sheppard felt that she did not need to be at a second formal meeting. The following comments from her and from R. Zach were communicated to B. Goodwin.

1. There is inconsistancy in the naming of related variables, such as CONCS8 and SO4MIX.
2. For a more thorough walkthrough, the whole process should start at the 'first' module, and work downwards to the 'last' module. This would catch problems such as changes in the definition and/or use and/or units in a variable used in different places.
3. Define UNSAT. ZONE in the data dictionary.
4. The existing data dictionary has an error in the definition of CMEAT: it is used in the DOSE module, not the DUST module.

WALKTHROUGH SUMMARY

REVIEW NUMBER 2 STARTING TIME

DATE 1986-Feb-18 ENDING TIME

PRODUCT IDENTIFICATION Figures 2 to 2.3 inclusive

PRODUCED BY Bruce Goodwin

BRIEF DESCRIPTION Radon, Dust, Airdis of SENES Code

MATERIALS USED IN THE REVIEW: *(check here ____ if supplementary list)*

IDENTIFICATION	DESCRIPTION
DFD's, DDD's	
Notes for the above	
Appendix B from SENES Phase II	
Appendix D from SENES Phase II	

PARTICIPANTS:

	NAME	SIGNATURE
LEADER	B. Goodwin	
RECORDER	T. Andres	
PRESENTER	B. Goodwin	
4.	B. Amiro	
5.		
6.		
7.		

APPRAISAL OF THE WORK UNIT:

ACCEPTED *(no further review)*	NOT ACCEPTED *(new review required)*
____ as is	____ major revisions
x with minor revisions	____ rebuild
	____ review not completed

SUPPLEMENTARY MATERIALS PRODUCED: DESCRIPTION AND/OR IDENTIFICATION

X COMMENT SHEETS

____ RELATED ISSUES LIST

____ OTHER

Walkthrough of the RADON and DUST Models [Review #2 (Pass #1)]

(Comments recorded by T.H. Andres Feb. 20, 1986)

1. BA Why consider only Ra in dust?

2. TA Numbers should be parameters

3. TA Unit inconsistency on 2.2×10^{-6}

4. TA FPREC DD- use "d" not days
   - should mention "non-winter"

5. TA FWIND1- "6 summer months"

6. TA EBKDFC shouldn't appear in fig. 2.2

7. BG RHOTAL not needed in PRADON

8. BG 1.? not known yet

9. TA RNFLUX - should have unit Bq $m^{-2}s^{-1}$

10. TA where did D go? - should be noted in Notes and in DD that a modified function is being used

11. TA Need better Note 2 to explain RHOTAL

12. TA Good idea to date verbal info from SENES

13. BG Is use of subscript I ok?

14. BA Pretty standard equations throughout

15. ALL Problems with units in RACON1, RACON2 should be fixed up

WALKTHROUGH SUMMARY

REVIEW NUMBER 3 (Pass #1) STARTING TIME 11:12

DATE Feb. 20/86 ENDING TIME 12:00

PRODUCT IDENTIFICATION NUTP Tailings Model

PRODUCED BY B. Goodwin

BRIEF DESCRIPTION LAKE and LAKE PH modules

MATERIALS USED IN THE REVIEW: *(check here ______ if supplementary list)*

IDENTIFICATION	DESCRIPTION
dfds 4 to 4.3.4 incl.	

PARTICIPANTS:

NAME	SIGNATURE
LEADER B. Goodwin	BW Goodwin
RECORDER I.H. Andres	Ingolf
PRESENTER B. Goodwin	
4. K. Hoffman	Karen Hoffman
5. B. Amiro	BD Amiro
6.	
7.	

APPRAISAL OF THE WORK UNIT:

ACCEPTED *(no further review)*	NOT ACCEPTED *(new review required)*
______ as is	x major revisions
______ with minor revisions	______ rebuild
	______ review not completed

SUPPLEMENTARY MATERIALS PRODUCED: DESCRIPTION AND/OR IDENTIFICATION

✓ COMMENT SHEETS

______ RELATED ISSUES LIST

______ OTHER

Walkthrough of the Lake Module [Review #3 (Pass 1)]

(Comments recorded by T.H. Andres, February 20, 1986)

Figure 4.0
1. BA BF should be going into 4.2, and should appear in PLAKE
2. TA PLAKE note should be on Figure 4.0
3. KH PHLAKE note is not needed.
4. TA How do we treat TIME and DELT?
5. TA Extend CLAKE arrow.

Figure 4.1
6. KH Extend XLAML arrow.
7. KH No note necessary.
8. BG Ambiguity in def'n of VLAKE in SENES report.
9. TA Figures 4.0 and 4.1 could be combined.

Figure 4.1.1
10. KH Missing unit check.
11. KH Missing reference to page in report.
12. TA MLAKE is misnamed; not mass but activity.
13. BG Should definition of symbols in SENES report continue? Yes. Should come before equations.

Figure 4.1.2
13. KH MSED shows up in top left, but not in upper diagram, and not used.
14. BA CLAKE and CSED should be MLAKE and MSED in notes 1 and 2 (3 places all told).
15. BA Need a sentence about BETA and XK2. Need a reference.
16. BG Units are missing and the check should be in.
17. TA Use "complicated", not "complex".
18. KH Note 2: equation for conversion should be given.
19. TA Term in DD "500-year variance" should be explained.
20. KH Diagram should be reorganized to better show XK2.
21. TA Does RAMIX = $\{Ra\}_{r,t}$? Used as if same, but def'ns are quite different.
22. TA XK1 DD says used in DOSE model; should be LAKE.
23. TA Equations, though complicated, should be here, so that it will be clear exactly what is being used.

Figure 4.1.3
24. KH Need to use "where QLAKE = ..." or put QLAKE first.
25. KH Need reference for QLAKE equation.

Figure 4.2
26. KH Need equation reference "p. E4".
27. BG CFISH def'n should include "average annual"
28. TA CFISH def'n should have "wet weight".
29. TA CLAKE def'n should have "average annual".

Figure 4.3
30. BA 4.3.1 should have ACTCO going in.
31. BA 4.3.4 should have CAMIX and SO4MIX.
32. KH 4.3.1, 4.3.2, 4.3.3 outputs should be named.
33. TA Drop TIME.
34. BG Extend arrows for XLAML and DELT.

Figure 4.3.1
35. BA Is there a need to show iteration in diagram? No.
36. KH Where did original guess "pH = 6" come from?
37. TA No unit check.
38. TA pH not defined: related to PHLAKE?
39. TA No reference to equation.

Figure 4.3.2
40. BA Equation needed.
41. KH Unit check needed.
42. BA TIME removed.
43. TA I didn't get one.

Figure 4.3.3
44-46. Same comments as for 4.3.2.
47. BA Wrong equation reference (3.2. not 2.3).

Figure 4.3.4
48. BG Unit inconsistency between CAMIX and constants (e.g. 10^{-4}).
49. KH Unit check missing.

WALKTHROUGH SUMMARY

REVIEW NUMBER 3 (~~third~~ second pass) STARTING TIME 11:30

DATE 21-Feb-1986 ENDING TIME 12:15

PRODUCT IDENTIFICATION NUTP Tailings Model

PRODUCED BY B. Goodwin

BRIEF DESCRIPTION LAKE and LAKE PH models

MATERIALS USED IN THE REVIEW: *(check here ______ if supplementary list)*

IDENTIFICATION	DESCRIPTION
dfd's, ddd's 4.0 to 4.3.4 incl.	Figures 4.0 to 4.3.4 incl.
data dictionary	
comments from first pass	
notes for dfds, ddd's	

PARTICIPANTS:

	NAME	SIGNATURE
LEADER	B. Goodwin	
RECORDER	B. Goodwin	
PRESENTER	B. Goodwin	
4.	B. Amiro	B Amiro
5.	K.J. Hoffman	Karen Hoffman
6.	T. Andres	
7.		

APPRAISAL OF THE WORK UNIT:

ACCEPTED *(no further review)*

______ as is

X with minor revisions

NOT ACCEPTED *(new review required)*

______ major revisions

______ rebuild

______ review not completed

SUPPLEMENTARY MATERIALS PRODUCED: DESCRIPTION AND/OR IDENTIFICATION

X COMMENT SHEETS

______ RELATED ISSUES LIST

______ OTHER

Note: the "notes for dfds ..." requires major revisions.

Walkthrough of the "Notes for DFDs etc" and the Lake Modules
[Review #3 (Pass 2)]

(Comments recorded by B.W. Goodwin, February 21, 1986)

Comments on the "notes for dfds etc."

THA 1. Title - should be "notes for specifications of utap-tailings model"
THA 2. There should be separate notes for dfds, ddds and the data dictionary
KJH 3. Leave out the word "logical" (note 1.)
THA 4. Use reference for note 1 on dfds and rewrite for clarity
THA 5. Note 2: replace with "when augmented KJH 6.Note 2: replace "similar to its parent..." with "equal to its parent...
THA 7. Note 2: use definition from Smith (CSAM) for ddd
THA 8. Note 5: remove "superscripted"
THA 9. Note 6: change "shown" to "represented by"
KJH 10. Note 6: remove "boxes"
KJH 11. Note 7: add that the ddds also include the dd definition of the variable
KJH 12. Note 7: the data definition should be removed from the ddd.
THA 13. Note 7: strange to define what a ddd includes here
KJH 14. Note 8: watch use of "parameter" and "variable"
THA 15. Note 9: should refer to the SENES code
KJH 16. Note 10: last word should be ddd not figure
THA 17. Note 10: use "significant proabability" or "significant chance"
KJH 18. Note 11. Add: if "CFISH" etc is in a bubble, then the next Figure expansion is a ddd, not a dfd

Comments on Figures 4 to 4.3.4

THA 1. Not enough problems to make the walkthrough enjoyable
THA 2. Names in bubbles do not match dd names.
KJH 3. Figure 4.1.1: unit check is wrong
KJH 4. Unit checks should follow the format of the equations
THA 5. Figure 4.1.1: fix "symbols" for CLAKE and MLAKE
KJH 6. Figure 4.1.2: arrived upside down
KJH 7. Figure 4.1.2: if MSED is referenced, are they the same ddd? If similar, what is the difference?
THA 8. Figure 4.1.2: MOLTOBxRAMIX does not yet have a refernced bubble number.
THA 9. Equation for XK2 should have "see note 1"
THA 10. Unit incompatability not included in the equations
THA 11. Data dictiorary out of order for MODULE and MSED
THA 12. Change order of equations and unit checks consistently
THA 13. Figure 4.2: names don't match
THA 14. CFISH dd entry needs fixing- units section is separated onto two lines

THA 15. Figure 4.3: names don't match

KJH 16. Figure 4.3.1: relevance of the initial guess not clear

THA 17. Figure 4.3.1: include note that the equation has been modified

THA 18. Figure 4.3.2 and .3: fix CALAKE (t=0); should be CALAKE(0)

KJH 19. Figure 4.3.4: Messy.

THA 20. Page references are wrong: C-6 should be C-5

THA 21. Equations are hard to read.

THA 22. Note 2 is wrong; the conversion factor is 10^{+3}

THA 23. Note that the equations are modified from the original SENES text

WALKTHROUGH SUMMARY

REVIEW NUMBER 4 (pass 1) STARTING TIME 10:30

DATE ENDING TIME 11:30

PRODUCT IDENTIFICATION Figures 1 to 1.3 incl.

PRODUCED BY T.H. Andres and B.W. Goodwin

BRIEF DESCRIPTION TSOURC module: for RACON1, CAMIX, SO4MIX

MATERIALS USED IN THE REVIEW: IDENTIFICATION	(check here ______ if supplementary list) DESCRIPTION
DFD's, DDD's	0.1, 1.1, 1.2, 1.2.1, 1.2.1.1,
Notes for the above	1.2.1.2, 1.2.1.3, 1.2.1.4, 1.2.2,
Appendix A from SENES	1.2.3, 1.2.3.1, 1.2.3.2, 1.2.3.3,
Data dictionary	1.2.3.4, 1.2.3.5, 1.2.3.6, 1.2.5
	1.3, 1.3.1, 1.3.2, 1.3.3, 1.3.4-1.3.5-1.3.6

PARTICIPANTS: NAME		SIGNATURE
LEADER	B. Goodwin	B Good
RECORDER	K. Hoffman	
PRESENTER	T. Andres & B. Goodwin	
4.	M. Stephens	M. Stephens
5.	T. Melnyk	T. Melnyk
6.	(R. Holmes)	
7.		

APPRAISAL OF THE WORK UNIT:

ACCEPTED (*no further review*)	NOT ACCEPTED (*new review required*)
______ as is	______ major revisions
* with minor revisions	______ rebuild
	X review not completed

SUPPLEMENTARY MATERIALS PRODUCED:	DESCRIPTION AND/OR IDENTIFICATION
X COMMENT SHEETS	
X RELATED ISSUES LIST	* on current sheets
______ OTHER	

Walkthrough of the TSOURC Modules [Review #4 (Pass 1)]

(Comments recorded by K.J. Hoffman February 19, 1986)

KH - too much for one walkthrough
KJ - summary says less than notice of meeting
MS - more overview of NUTP model would be helpful
TM,BG - dd could have GYPSUM - free for DEPTH1
MS * is Gypsum $CaSO_4$ or $CaSO_4$ with 2 waters
BG - numbering is not all updated
TM - didn't get copy of covering notes for dfd's etc.
KH - names on diverging data flows should be before the divergence
BG - is busy
TM - didn't get overview diagram either
BG - PSOURC not described on fig. 1.1
BG - name of 1.2 misleading - gypsum related to radium - TA response
KH - repeating definitions on diagrams and dd is redundant and error-prone -
TA - definition on DDD not yet matched to dd
TM - inconsistency of units eg. GYPBLK and GYPLFT -
MS * X, X mass-fraction, mole-fraction also inconsistent -

fig. 1.1

TA - unit check on 1.1 136 has units g/mol
TM * 136 right - gypsum with or without H_2O?
KH - need $\frac{1}{136}$ as an input to ddd?
KH - unit check is not trivial

fig. 1.2

KH - general 1/0 checks - no time for rest
BG - PM1XZ and PCHEM not defined on diagram
BG,TA - PM1XZ shouldn't have contents of CHMCNS
BG,TA - how to show time dependency?
need consistency across all dfds

1.2.2

KH - r is input, missing from DDD
TA,BG - (is from CSINIT)
KH - where does initial time guess come from
KH,TA - unit check is where?, should be here
TA - A.3.21 was modified, diagram should say so

1.2.3.3

TM - equation only valid for T > TFINAL
discontinuity at TFINAL
appears to be only used for > TFINAL, so might not cause errors,
but should be shown
TA - put down unit check

1.2.3.4

TM - need T > TFINAL on 2nd equation
TM - definition of MGYPI first or referenced
"where MGYPI = ..."

1.2.3.5, 1.2.3.6

TM - block TIME on diagram

1.3

KH - CTH230 on 1.3 but not on 1
BG - lines on RAINTH & THFRAC not long enough

1.3.1

TM - what is 1.196×10^{-13}

1.3.4,5,6

MS * p. A24 4th line equation 5.6 not readable, equation has an unmatched parenthesis -
TH-230 not defined

MS - DDD seems ok

KH - diagrams should be separated?

BG - #1.3.5 missing

MS - half lives TH230, TH232 should always be distinguished, it both are used (TM-dd has it)

TA - waste other time if one person focuses on a given section

BG - disagrees

1.2.1.1

TA - SOLPRD doesn't match SOLPRO on diagram

MS - α not defined p. A6, equation 3.3 -

WALKTHROUGH SUMMARY

REVIEW NUMBER 4 (Pass 2) STARTING TIME --

DATE Feb.25 - March 3 ENDING TIME --

PRODUCT IDENTIFICATION NUTP tailings model

PRODUCED BY B. Goodwin and T. Andres

BRIEF DESCRIPTION TSOURC modules

MATERIALS USED IN THE REVIEW: *(check here _____ if supplementary list)*

IDENTIFICATION	DESCRIPTION
DFD's DDD's	1.0 to 1.6 incl.
Notes for above	
data dictionary	
Appendix A from SENES	

PARTICIPANTS:

	NAME	SIGNATURE
LEADER		
RECORDER	B. Goodwin	
PRESENTER		
4.	M. Stephens	comments from each were gathered
5.	T. Melnyk	separately to facilitate the
6.	K. Hoffman	completion of this pass of the
7.		walkthrough.

APPRAISAL OF THE WORK UNIT:

ACCEPTED *(no further review)*
_____ as is
_____ with minor revisions

NOT ACCEPTED *(new review required)*
_____ major revisions
_____ rebuild
X review not completed

SUPPLEMENTARY MATERIALS PRODUCED: DESCRIPTION AND/OR IDENTIFICATION
X COMMENT SHEETS
_____ RELATED ISSUES LIST
_____ OTHER

Walkthrough of the TSOURC Modules [Review #4 (Pass 2)]

(Comments recorded by B.W. Goodwin, February 25 to March 3, 1986)

Comments from M.E. Stephens (February 25, 1986)

1. Lists of parameters in the clusters should appear on the dfds in which they first appear and in the first dfd (or ddd) immediately below to facilitate cross-checking.
2. Many variables are time-dependent. Useful to distinguish somehow which are time-dependent.
3. Figure 1.2: is TFINAL required in Fig. 1.2.5?
4. Figure 1.2.1: bubble 1.2.1.3 title should be ...RATE CONSTANT... not ...RATE... .
5. Figure 1.2.3: first bubble (?) to use TIME. How is the need for TIME shown?
6. Figure 1.2.3.1: what happened to the variables TIME and Z (position)? Where is the list of symbols? These are very helpful in cross-checking and in checking equations used. In some cases, different symbols are used in different equations.
7. Figure 1.2.3.2: list of symbols?
8. Figure 1.2.3.3: should there be a "-" in the second equation given for ION? List of symbols?
9. Figure 1.2.3.4: list of symbols? TIME is needed again. Not clear what the initial condition is for GYPLFT.
10. Figure 1.2.3.5: list of symbols?
11. Figure 1.2.3.6: list of symbols?
12. Figure 1.2.4: DEPTH1 should also be shown as an 'output' variable, since it is used elsewhere.
13. Figure 1.2.4.1: Not clear whether DEPTH1 is an increasing positive quantity or a decreasing negative quantity. Add the following ....iteratively for DEPTH1:. Check that the third term has the right (-) sign. Equation number referenced should be (A.3.20). Note that DEPTH1 is required at time=TIME and TIME+DELT for use in the next figure.
14. Figure 1.2.4.2: TIME? List of symbols? Is DELH a negative or positve quantity (cf. DEPTH1).
15. Figure 1.2.4.3: TIME? List of symbols? Not clear if the the oxidizing zone (depth = DUNSAT) is identical to the unsaturated zone (depth = 1 m).
16. Figure 1.2.5: TIME? DUNSAT and TFINAL are listed as required parameters, but are not used in the equations given.
17. Figure 1.3.5: TIME? Caption "RATH23" missing from the ellipse. Also enter TIME=t in the list of symbols.
18. Figure 1.3.6: TIME? Also enter TIME=t in the list of symbols.
19. Figure 1.3.7.1: where does the constant 3.788×10^{-4} come from?
20. Figure 1.3.7.4: list of symbols?

21. Figure 1.4: should note somewhere that m(), x() and $C_r()$ are all required at TIME and TIME+DELT. CRAGIN is defined differently here than in eq. (A.2.2). Are the two expressions always equivalent?
22. Figure 1.5.1: TIME? List of symbols?
23. Figure 1.5.2: list of symbols? Note somewhere that RAMIX is required at TIME to calculate RAMIX at TIME+DELT.
24. Figure 1.6: TIME? Check the list of symbols: is the expression for DEPTH2 correct?

Comments from K.J. Hoffman (February 28, 1986)

1. Overlapping definitions are difficult to follow; for example PNONRA = DUNSAT+NPRCP+PCHEM+PGYP+PMIXZ. but PGYP=NPRCP+DUNSAT+DELT. So why not just put DELT in PNONRA and omit PGYP from PNONRA? Also you could omit DELT in PMIXZ.
2. In general, unit checks should have the same format as the equations; ie. explicity show which units come from the denominator and which come from the numerator, etc. Need to show the "+" between operations in some cases.
3. Leave equations out of the data dictionary - some are anyway.
4. References back to SENES should give a page number if their equation isn't numbered.
5. Figure 1.0: CTH230 is not shown in Fig. 0 as an output variable. PRAMIX is not in the data dictionary.
6. Figure 1.2.1.1: inconsistent use of the symbol K'.
7. Figure 1.2.1.2: reference page equation is found on.
8. Figure 1.2.1.3: PYRITE is missing from the data dictionary.
9. Figure 1.2.1.4: reference page equation is found on. Complete unit check.
10. Figure 1.2.2: bracket missing on unit check. When was h set to DUNSAT? Confirmed by whom?
11. Figure 1.2.3.2: Unit check not complete; second term is missing. Is the J cutoff okay in equation (A.3.9)? Why talk about PHUNST here? Give the symbol used in SENES for CONCS8.
12. Figure 1.2.3.3: Error in the expression here compared with equation (A.7.2): a "-" is missing. Is there a new symbol used in SENES (or here?) for CONCS8?
13. Figure 1.2.3.4: Second expression for GYPLFT: should the second term be "+NPRCP....."? How were the equations modified for clarity?
14. Figure 1.2.3.5: Add T>TFINAL to the second equation.
15. Figure 1.2.3.6: Add T>TFINAL to the second equation.
16. Figure 1.2.4: DEPTH1 in Figure 1.2 is shown exiting as an output variable, but not here.
17. Figure 1.2.4.2: Who is responsible for the second equation?
18. Figure 1.2.4.3: Where is the second equation from?
19. Figure 1.2.5: The unit check is not obvious. Check for consistent use of 'r' and 'R' here and in Figure 1.2.2.

20. Figure 1.3: PRAFTH is missing on Figure 1.0, and is not in the dd. THLAM and THFRAC are in PRAFTH, but are not in Figure 1.
21. Figure 1.3.2: Page A-3 is a better reference to the equation. RAINGP is not in the dd.
22. Figure 1.3.3: Unit check should be ...a^{-1} / a^{-1}.
23. Figure 1.3.4: Use Page A-3 as the reference. The unit check is incomplete.
24. Figure 1.3.5: RAINTH is not in the dd. Unit check should also consider the EXP terms. In the second note, who is assuming that the two symbols for CTH230 are equivalent?
25. Figure 1.3.6: Messy. Unit check is missing and should not be stated as 'trivial'.
26. Figure 1.3.7.3: Unit check is 'trivial' again.
27. Figure 1.3.7.4: Unit check is 'trivial'.
28. Figure 1.3.7.5: RAINTH is not in the dd.
29. Data dictionary: What does ACT. abbreviate for in CONCRA?
30. Data dictionary: DEPTH1 uses equation (A.3.20).
31. Data dictionary: Modify the definition of GYPBLK to "....content in bulk tailings"?
32. Data dictionary: KPRIME definition uses "O" and "0" interchangeably. What are the four parameters in the lognormal distribution for KPRIME?
33. Data dictionary: Check for typo for SULPRD in PCHEM.
34. Data dictionary: What are the four parameters for in the normal distribution for NPRCP?
35. Data dictionary: What is PGYP? Does PMIXZ include AEQBM, CSINIT and KRATE? Don't like the overlap stated in PNONRA. Be more specific in PRAUP (the words 'some' and 'including' are too vague).
36. Data dictionary: RACON1: why not convert it here, once, instead of twice later?
37. Data dictionary: RALAM: fix type in the short definition. The value given for RALAM does not exist in the SENES parameter tables.
38. Data dictionary: RHOTAL: what are the four numbers in the normal distribution used for?
39. Data dictionary: SO4MIX: the equations given are confusing with regards to the conditions (TIME>TFINAL and...).
40. Data dictionary: THLAM: Value given does not exist in the SENES parameter tables.
41. Data dictionary: THSOLD: is (5.2) the correct equation number?

Comments from T.W. Melnyk (March 3, 1986)

1. Figure 1.1: replace "GYPSUM" with $CASO_4$ for clarity/preciseness. Use a variable for the constant.
2. Figure 1.2.1.4: Unit check is incomplete.
3. Figure 1.2.3.3: is the concentration per unit volume of tailings or per unit volume of porewater in the tailings? (I suspect the latter; also fix the dd entry for ION).

4. Figure 1.2.3.4: in dd entry for GYPLFT replace "..including.." with "..taking into account..."
5. Figure 1.2.3.6: add T>TFINAL to third equation. Obvious if you understand the model, but required by the programmers to keep the IF THEN ELSEs correct.
6. Figure 1.3.1: the constant appears to be equivalent to 1.0/MOLTOB, but 1/MOLTOB = 1.212×10^{-13}, not 1.196×10^{-13}. Which is correct?
7. Figure 1.5: PRAMIX in Figure 1.0 is not correctly referenced (with respect to its contents) here.
8. Figure 1.5.1: in the definitions specify explicitly whether concentrations are in the purewater or aqueous phase.
9. Figure 1.5.2: equation is not that in the SENES report, although it is noted that the equation given here is correct. I had to assume this equation was correct.
10. General comment: why does SENES give FD equations, as opposed to giving the originating differential equations, and why don't they use a standard ode solution, such as Runge-Kutta?
11. Data dictionary: CALIN - which "finite difference equation"? What are the constants given? Define them here.
12. Data dictionary: CONCS8 - add "...in the aqueous phase..." to the definition. Use "z" and "Z" consistently.
13. Data dictionary: CSINIT - same comment as above.
14. Data dictionary: CRAGIN - add "....bulk concentration...".
15. Data dictionary: EBKDFC - what are the constants?
16. Data dictionary: MOLTOB - see the comment above (#6.).
17. Data dictionary: NPRCP - "excluding" evapotranspiration? Better to use "accounting for..." or whatever is correct.
18. Data dictionary: SO4MIX - equations are calculated at Z=DUNSAT for CONCS8.
19. Data dictionary: SULIN - same comments as for CALIN. Add "aqueous phase" or equivalent.
20. Data dictionary: TIME - is "parameters" correct or should it be "variables"?
21. Data dictionary: TVOID - units should be [-].
22. Data dictionary: UO2TON and UPERTH: inconsistency in the manner in which these two are named and dimensioned.
23. Data dictionary: XK1 - should be "LAKE" not "DOSE" module.
24. Data dictionary: Z - insert "Vertical position ....". Strange to say it is an independent parameter that is not required! End the definition after the first sentence?

APPENDIX E. SAMPLE OUTPUT FROM SOFTOOL'S PROGRAMMING ENVIRONMENT AUDITOR

Softool's Programming Environment is a software development package aimed at the development of high-quality computer software. The package includes the following products:

AUDITOR: A FORTRAN auditor that accepts as input FORTRAN code and that produces documentation describing:
- deviations from user-defined standards;
- poor programming practices; and
- non-portable code.

AUDITOR adheres rigidly to the American National Standards Institute (ANSI) definition of FORTRAN. It is declared to be a powerful error detector, as can be verified by the results presented here. AUDITOR also flags any non-standard FORTRAN statements, so that a coded passed by AUDITOR should be acceptable, without modification, on any computer with an ANSI-compatable FORTRAN compiler.

INSTRUMENTER: An integrated collection of tools that is used to trace and test code, and to idenify key sections of code that would best benefit from code optomization.

DOCUMENTER A: A documentation package that accepts as input source code and DOCUMENTER A control statements. This tool provides the means to adopt a uniform documentation standard during the development of FORTRAN code.

The following pages contain sample output from AUDITOR, for successive passes of the subroutine BISECT. In the first pass (starting on page E-2), a non-standard documentation was used for BISECT, and some statements that could cause problems were noted. In the second pass (starting on page E-10), a number of errors arose because the programmer did not follow the user-defined standards for DOUBLE PRECISION statements, and many spurious errors resulted. These errors disappeared in the third and final pass (starting on page E-18).

AUDITOR produces three types of error messages: those that relate to ANSI standards, those that relate to code portability, and miscellaneous documentation. It is not necessary to correct all "errors" and "warnings"; however all messages should be formally addressed. In the third pass of BISECT, for example, the four ANSI standards messages, the 14 portability messages, and the eight miscellaneous messages were evaluated, and it was decided that no code changes were required. This procedure was adhered to for all runs using AUDITOR. Listings of all documentation provided by AUDITOR is available at the Whiteshell Nuclear Research Establishment, for all code developed for the uranium tailings model described in this report.

```
***********************************************************************
*                                                                     *
*                 S O F T O O L   C O R P O R A T I O N               *
*                                                                     *
*  -----------------------------------------------------------------  *
*                                                                     *
* F O R T R A N   A N A L Y S I S   A N D   D O C U M E N T A T I O N *
*                                                                     *
*                                                                     *
***********************************************************************
```

COPYRIGHT (C) 1982 SOFTOOL CORPORATION.
ALL RIGHTS RESERVED.
LICENSED MATERIALS. PROPERTY OF SOFTOOL CORPORATION.

SOFTOOL PRODUCT : BDCA1-01.01-[illegible]

```
***************************************************************************
DETCOL                FIRST PASS                MARCH 12, 1986  16:00
***************************************************************************
```

```
      SUBROUTINE BISECT (A, B, ACCUR, FUNC, ROOT, IERR)
C.......................................................................
C  FINDS A ROOT OF A FUNCTION BETWEEN TWO SPECIFIED LIMITS, USING
C  A BISECTION ALGORITHM.  BASED ON THE DESCRIPTION BY R.W. HAMMING
C  IN 'INTRODUCTION TO APPLIED NUMERICAL ANALYSIS', MCGRAW-HILL,
C  1971, CHPT. 2.
C
C  VERSION 1     FEB. 25, 1986     D.GOODWIN
C  VERSION 2     MAR. 04, 1986     D.GOODWIN
C.......................................................................
C
C  ARGUMENTS ARE AS FOLLOWS:
C      A, B    - INCLUSIVE LIMITS WITHIN WHICH IT IS ASSUMED THAT ONE
C                AND ONLY ONE ROOT IS LOCATED.
C      ACCUR   - DESIRED ACCURACY IN EACH ROOT.  IF ACCUR .LT. 1.0D-12,
C                IT WILL DEFAULT TO 1.0D-5
C      FUNC    - EXTERNAL FUNCTION TO BE SCANNED.  CALLS TO FUNC
C                ARE OF THE FORM:  Z = FUNC (X) WHERE X IS THE
C                INDEPENDENT VARIABLE
C      ROOT    - SCALAR CONTAINING THE ROOT FOUND.
C      IERR    - RETURNS AN ERROR CODE:
C                    IERR = 0  - NO ERRORS DETECTED, ROOT FOUND
C                    IERR = 1  - INTERVAL CONTAINS NO ROOTS OR AN
C                                EVEN NUMBER OF ROOTS.  NONE HAVE BEEN
C                                FOUND: RE-USE BISECT WITH A SMALLER
C                                RANGE.
C                    IERR = 2  - A AND B ARE IDENTICAL, AND ARE NOT A
C                                ROOT.
C                    IERR = 3  - NO ROOTS FOUND, BUT A POLE EXISTS WITHIN
C                                THIS RANGE.
C.......................................................................
      IMPLICIT LOGICAL (A-Z)
      INTEGER IERR
      INTEGER J
      INTEGER NOTRY
      DOUBLE PRECISION A
      DOUBLE PRECISION ACCUR
      DOUBLE PRECISION B
      DOUBLE PRECISION F1
      DOUBLE PRECISION F2
      DOUBLE PRECISION F3
      DOUBLE PRECISION FUNC
      DOUBLE PRECISION ROOT
      DOUBLE PRECISION X1
      DOUBLE PRECISION X2
      DOUBLE PRECISION X3
      IF (ACCUR .LT. 1.0D-12) ACCUR = 1.0D-5
C  ......ARE A AND B IDENTICAL AND AT A ROOT?
      IF (DABS (A-B) .EQ. 0.0D00) THEN
         IF (FUNC(A) .EQ. 0.0D00) THEN
            IERR = 0
            ROOT = A
         ELSE
```

```
            IERR = 2
         ENDIF
      ELSE
C  ......SEARCH FOR A ROOT BETWEEN A AND B INCLUSIVE.
         X1 = A
         X2 = B
         F2 = FUNC(X2)
         F1 = FUNC(X1)
C  .........AT LEAST ONE EXISTS IF FUNC(A) AND FUNC(B) HAVE OPPOSITE
C           SIGNS.  IF THEY HAVE THE SAME SIGN, THERE COULD BE ZERO
C           OR AN EVEN NUMBER OF ROOTS - AN ERROR CONDITION.
         IF (F1*F2 .GT. 0.0D00) THEN
            IERR=1
C  .........IS THERE A ROOT AT EITHER EXTREME?
         ELSE IF (F1 .EQ. 0.0D00) THEN
            IERR = 0
            ROOT = X1
         ELSE IF (F2 .EQ. 0.0D00) THEN
            IERR = 0
            ROOT = X2
         ELSE
C  ................EACH SEARCH WILL USE UP TO NOTRY BISECTIONS, WHICH
C                  WILL GUARANTEE THE REQUIRED ACCURACY.  THE CONSTANT
C                  BELOW IS ALOG(2)
            NOTRY = INT ( (DLOG(DABS(B-A)) - DLOG(ACCUR) )
     1                / 0.6931471806D00 )
            DO 330 J = 1, NOTRY
               X3 = 0.5D00 * (X1 + X2)
               F3 = FUNC (X3)
               IF ( (F1 * F3) .GT. 0.0D00) THEN
C  ................ROOT COULD BE BETWEEN X3 AND X2
                  X1 = X3
                  F1 = F3
               ELSE IF ( (F1 * F3) .LT. 0.0D00) THEN
C  ................ROOT COULD BE BETWEEN X3 AND X1
                  X2 = X3
                  F2 = F3
               ELSE
C  ................FOUND A ROOT BANG ON; SAVE IT
                  IERR = 0
                  ROOT = X3
                  GO TO 350
               ENDIF
  330       CONTINUE
C  .............SOMETHING HAS NOW BEEN LOCATED WITHING THE DESIRED
C                  ACCURACY.  IT IS ASSUMED TO BE A ROOT IF F3 LIES
C                  BETWEEN F1 AND F2; OTHERWISE IT IS (PROBABLY) A
C                  POLE.  CALCULATE A NEW X3 AND F3 AND TEST.
            X3 = 0.5D00 * (X1 + X2)
            F3 = FUNC(X3)
            IF ( (F3-F2)*(F1-F3) .GT. 0.0D00) THEN
               IERR = 0
               ROOT = X3
            ELSE
               IERR = 3
            ENDIF
         ENDIF
      ENDIF
  350 CONTINUE
C           WRITE (6,3333) IERR,X1,X2,X3,F1,F2,F3
```

```
      C 3333    FORMAT (1H , 'BISECT IERR=',I3,/5X,'X1-X3,F1-F3:',/6E12.3)
                RETURN
                END
```

```
*********************************************************************
SOFTOOL
*********************************************************************


*********************************************************************
*                                                                   *
*     F O R T R A N    S T A N D A R D S    D O C U M E N T A T I O N   *
*                                                                   *
*********************************************************************
*        *                                                 *        *
*  STMT  *                     MESSAGE                     * KEY TO *
* NUMBER *                                                 * MANUAL *
*        *                                                 *        *
*********************************************************************
*        *                                                 *        *
*    40  *  ILLEGAL COMBINATION OF TYPES IN ADDITION OR    *        *
*        *  SUBTRACTION                                    *  E128  *
*        *                                                 *        *
* ------ * ----------------------------------------------- * ------ *
*        *                                                 *        *
*  ....  *  SUBPROGRAM ALTERS DUMMY ARGUMENT ACCUR         *  W016  *
*        *                                                 *        *
*  ....  *  SUBPROGRAM ALTERS DUMMY ARGUMENT ROOT          *  W016  *
*        *                                                 *        *
*  ....  *  SUBPROGRAM ALTERS DUMMY ARGUMENT IERR          *  W016  *
*        *                                                 *        *
* ------ * ----------------------------------------------- * ------ *
*        *                                                 *        *
*  ....  *  STANDARDS COMPLIANCE INDEX                     *        *
*        *  AVERAGE NUMBER OF ERRORS PER STATEMENT   0.015 *        *
*        *  AVERAGE NUMBER OF WARNINGS PER STATEMENT 0.044 *  C004  *
*        *                                                 *        *
*********************************************************************
```

```
*************************************************************************
SOFTOOL
*************************************************************************


*************************************************************************
*                                                                       *
*  F O R T R A N   P O R T A B I L I T Y   D O C U M E N T A T I O N    *
*                                                                       *
*************************************************************************
*        *                                                 *            *
* STMT   *                   MESSAGE                       * KEY TO     *
* NUMBER *                                                 * MANUAL     *
*        *                                                 *            *
*************************************************************************
*        *                                                 *            *
*    17  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    19  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    20  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    31  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    33  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    35  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    44  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    47  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*    53  *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM *           *
*        *  DEPENDENT                                      *   P015     *
*        *                                                 *            *
*  ....  *  IDENTIFIER DABS MATCHES THE NAME OF AN INTRINSIC *           *
*        *  FUNCTION                                       *   P022     *
*        *                                                 *            *
*  ....  *  IDENTIFIER INT MATCHES THE NAME OF AN INTRINSIC *            *
*        *  FUNCTION                                       *   P022     *
*        *                                                 *            *
*  ....  *  NAME DABS MATCHING NON-GENERIC INTRINSIC       *            *
*        *  FUNCTION NAME SHOULD HAVE EXPLICIT TYPING      *   P025     *
*        *                                                 *            *
*  ....  *  NAME INT MATCHING NON-GENERIC INTRINSIC FUNCTION *          *
*        *  NAME SHOULD HAVE EXPLICIT TYPING               *   P025     *
*        *                                                 *            *
*  ....  *  INT IS SYSTEM DEPENDENT                        *   P027     *
*        *                                                 *            *
* ------ * ----------------------------------------------- * ------     *
*        *                                                 *            *
*   ....  *  PORTABILITY INDEX                                       *          *
*         *  AVERAGE NUMBER OF PORTABILITY PROBLEMS                  *          *
*         *  PER STATEMENT      0.206                                *   C006   *
*         *                                                          *          *
**************************************************************************
```

```
*****************************************************************
DETCOL
*****************************************************************


*****************************************************************
*                                                               *
*  F O R T R A N   A D D I T I O N A L   D O C U M E N T A T I O N   *
*                                                               *
*****************************************************************
*         *                                            *         *
* STMT    *                 MESSAGE                    * KEY TO  *
* NUMBER  *                                            * MANUAL  *
*         *                                            *         *
*****************************************************************
*         *                                            *         *
*  ....   *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE          *         *
*         *  DEFINED BUT NOT REFERENCED:               *  D021   *
*         *  J                                         *         *
*         *                                            *         *
*  ....   *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE          *         *
*         *  REFERENCED BUT NOT DEFINED BY THIS PROGRAM UNIT:  *  D023   *
*         *  A       B                                 *         *
*         *                                            *         *
*  ....   *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE          *         *
*         *  DEFINED BUT NOT REFERENCED BY THIS PROGRAM UNIT:  *  D024   *
*         *  ROOT    ITER                              *         *
*         *                                            *         *
*  ....   *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE EACH *         *
*         *  DEFINED IN ONLY ONE STATEMENT:            *  D025   *
*         *  ACCUR   J                                 *         *
*         *                                            *         *
*  ....   *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE EACH *         *
*         *  DEFINED IN MORE THAN ONE STATEMENT:       *  D026   *
*         *  ROOT    ITER    F1      F2      F3      X1 *         *
*         *  X2      X3                                *         *
*         *                                            *         *
*  ....   *  THE FOLLOWING IDENTIFIERS ARE DECLARED BY  *         *
*         *  DEFAULT:                                  *  D027   *
*         *  DABS    INT     DLOG                      *         *
*         *                                            *         *
*  ....   *  THE FOLLOWING IDENTIFIERS WERE ACTUAL ARGUMENTS  *         *
*         *  IN A FUNCTION OR SUBROUTINE CALL:         *  D031   *
*         *  A       ACCUR   X1      X2      X3      DABS *      *
*         *                                            *         *
*  ....   *  THE FOLLOWING IDENTIFIERS WERE REFERENCED BUT    *         *
*         *  NOT DEFINED:                              *  D051   *
*         *  NCTRY                                     *         *
*         *                                            *         *
*---------*--------------------------------------------*---------*
*         *                                            *         *
*  ....   *  DOCUMENTATION OPTION INDEX                *         *
*         *  AVERAGE NUMBER OF DOCUMENTATION OPTION    *         *
*         *  MESSAGES PER STATEMENT    0.118           *  C008   *
*         *                                            *         *
*****************************************************************
```

```
$13$DUA4:[PROJECT.SYVAC.SYVAC.SC.PEDT]BISECT.LIS;1                                    18-M R-

*****************************************************************************
CFTCCL                     SECOND PASS      MARCH 18, 1986    9:06
*****************************************************************************

                           (DID NOT  FOLLOW  USER-DEFINED  SYNTAX
                            STANDARDS  FOR  DOUBLE PRECISION  STATEMENTS
                            - ALL OF WHICH HAVE  BEEN  DELETED)
   1         SUBROUTINE BISECT (A, B, ACCUR, FUNC, ROOT, IERR)
       C
       C******************************************************************** BISECT
       C  FINDS THE ROOT OF A FUNCTION BETWEEN TWO SPECIFIED LIMITS, USING
       C  A BISECTION ALGORITHM.  BASED ON THE DESCRIPTION BY R.W. HAMMING
       C  IN 'INTRODUCTION TO APPLIED NUMERICAL ANALYSIS', MCGRAW-HILL,
       C  1971, CHPT. 2.
       C
       C  86-FEB-25  VERSION 01  R.GOODWIN
       C  86-MAR-04  VERSION 02  R.GOODWIN
       C  86-MAR-12  VERSION 03  R.GOODWIN
       C
       C*********************************************************************
       C
   2         IMPLICIT LOGICAL (L-L)
       C                      ! limits defining a range within which a
       C                        root is located [X]
       C                      ! relative accuracy desired in the root.  IERR is set
       C                        if ACCUR is less that 1.0D-12 [X/X]
       C                      ! external function to be scanned.  Calls to FUNC
       C                        are of the form:  Z = FUNC (X) where X is the
       C                        independent variable.  [Z]
       C                      ! contains the root found [X]
   3         INTEGER IERR
       C                      ! error code with the following values & meaning []:
       C                        IERR = 0  - no errors detected, root found
       C                        IERR = 1  - interval contains no roots or an even
       C                                    number of roots.  None have been found;
       C                                    re-use bisect with a smaller range.
       C                        IERR = 2  - A and B are identical and are not a
       C                                    root.
       C                        IERR = 3  - no roots found, but a pole has
       C                                    has probably been found.
       C                        IERR = 4  - specified accuracy is less that 1.0D-12
   4         INTEGER J
       C                      ! loop index []
   5         INTEGER NOTRY
       C                      ! number of bisections required to acheive the
       C                        specified accuracy. []
       C                      ! temporary variables used to store values of FUNC
       C                        evaluated at X1, X2 and X3.  [Z]
       C                      ! temporary variables used to store values of the
       C                        independent variable required by FUNC.  [X]
       C                      ! set equal to 0.0D00 for comparisons [X]
   6         DATA ZERO /0.0D00/
       C  ......CHECK FOR 'REASONABLE' ACCURACY
   7         IF (ACCUR .LT. 1.0D-12) THEN
   8            IERR = 4
   9         ELSE IF ( ABS(A-B) .EQ. ZERO) THEN
       C  ......ARE A AND B IDENTICAL AND AT A ROOT?
  10            IF (FUNC(A) .EQ. ZERO) THEN
```

```
                 IERR = 0
                 ROOT = A
              ELSE
                 IERR = 2
              ENDIF
           ELSE
C   .......SEARCH FOR A ROOT BETWEEN A AND B INCLUSIVE.
              X1 = A
              X2 = B
              F2 = FUNC(X2)
              F1 = FUNC(X1)
C   ..........AT LEAST ONE EXISTS IF FUNC(A) AND FUNC(B) HAVE OPPOSITE
C                SIGNS.  IF THEY HAVE THE SAME SIGN, THERE COULD BE ZERO
C                OR AN EVEN NUMBER OF ROOTS - AN ERROR CONDITION.
              IF (F1*F2 .GT. ZERO) THEN
                 IERR=1
C   ..........IS THERE A ROOT AT EITHER EXTREME?
              ELSE IF (F1 .EQ. ZERO) THEN
                 IERR = 0
                 ROOT = X1
              ELSE IF (F2 .EQ. ZERO) THEN
                 IERR = 0
                 ROOT = X2
              ELSE
C   ................EACH SEARCH WILL DO UP TO NTRY BISECTIONS, WHICH
C                    WILL GUARANTEE THE REQUIRED ACCURACY.  THE CONSTANT
C                    BELOW IS ALOG(2)
                 NTRY = INT ( ( LOG( ABS(B-A) ) - LOG(ACCUR) )
     1                 / 0.69314718056 )
                 DO 330 J = 1, NTRY
                    X3 = 0.5D00 * (X1 + X2)
                    F3 = FUNC (X3)
                    IF ( (F1 * F3) .GT. ZERO) THEN
C   ................ROOT COULD BE BETWEEN X3 AND X2
                       X1 = X3
                       F1 = F3
                    ELSE IF ( (F1 * F3) .LT. ZERO) THEN
C   ................ROOT COULD BE BETWEEN X3 AND X1
                       X2 = X3
                       F2 = F3
                    ELSE
C   ................FOUND A ROOT BANG ON; SAVE IT
                       IERR = 0
                       ROOT = X3
                       GO TO 350
                    ENDIF
  330            CONTINUE
C   .............SOMETHING HAS NOW BEEN LOCATED WITHING THE DESIRED
C                    ACCURACY.  IT IS ASSUMED TO BE A ROOT IF F3 LIES
C                    BETWEEN F1 AND F2; OTHERWISE IT IS (PROBABLY) A
C                    POLE.  CALCULATE A NEW X3 AND F3 AND TEST.
                 X3 = 0.5D00 * (X1 + X2)
                 F3 = FUNC(X3)
                 IF ( (F3-F2)*(F1-F3) .GT. ZERO) THEN
                    IERR = 0
                    ROOT = X3
                 ELSE
                    IERR = 3
                 ENDIF
              ENDIF
          ENDIF
    990 CONTINUE
C             WRITE (6,3333) IERR,X1,X2,X3,F1,F2,F3
C 3333        FORMAT (1H , 'BISECT IERR=',I3,/5X,'X1-X3,F1-F3:',/6E12.3)
          RETURN
          END
```

```
_T133$DUA4:[PROJECT.SYVAC.SYVAC.SC.PRDTBISECT.LIS;1                                18-MAR-

*******************************************************************
SC TOOL
*******************************************************************


*******************************************************************
*                                                                 *
*   F O R T R A N    S T A N D A R D S    D O C U M E N T A T I O N   *
*                                                                 *
*******************************************************************
*        *                                              *          *
*  STMT  *                   MESSAGE                    * KEY TO   *
*  NUMBER *                                             * MANUAL   *
*        *                                              *          *
*******************************************************************
*        *                                              *          *
*     6  *  TYPE MISMATCH BETWEEN ITEM AND ASSOCIATED   *          *
*        *  CONSTANT                                    *   E043   *
*        *                                              *          *
*     7  *  ILLEGAL COMBINATION OF TYPES IN RELATIONAL  *          *
*        *  OPERATION                                   *   E130   *
*        *                                              *          *
*     9  *  ILLEGAL COMBINATION OF TYPES IN ADDITION OR *          *
*        *  SUBTRACTION                                 *   E128   *
*        *                                              *          *
*    10  *  ILLEGAL COMBINATION OF TYPES IN RELATIONAL  *          *
*        *  OPERATION                                   *   E130   *
*        *                                              *          *
*    13  *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A       *          *
*        *  CORRESPONDING BLOCK IF STATEMENT            *   E324   *
*        *                                              *          *
*    15  *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A       *          *
*        *  CORRESPONDING BLOCK IF STATEMENT            *   E324   *
*        *                                              *          *
*    16  *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A       *          *
*        *  CORRESPONDING BLOCK IF STATEMENT            *   E324   *
*        *                                              *          *
*    21  *  ILLEGAL COMBINATION OF TYPES IN MULTIPLICATION *   E120   *
*        *                                              *          *
*    23  *  ILLEGAL COMBINATION OF TYPES IN RELATIONAL  *          *
*        *  OPERATION                                   *   E130   *
*        *                                              *          *
*    26  *  ILLEGAL COMBINATION OF TYPES IN RELATIONAL  *          *
*        *  OPERATION                                   *   E130   *
*        *                                              *          *
*    29  *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A       *          *
*        *  CORRESPONDING BLOCK IF STATEMENT            *   E324   *
*        *                                              *          *
*    30  *  ILLEGAL COMBINATION OF TYPES IN ADDITION OR *          *
*        *  SUBTRACTION                                 *   E128   *
*        *                                              *          *
*    32  *  ILLEGAL COMBINATION OF TYPES IN ADDITION OR *          *
*        *  SUBTRACTION                                 *   E128   *
*        *                                              *          *
*    34  *  ILLEGAL COMBINATION OF TYPES IN MULTIPLICATION *   E120   *
*        *                                              *          *
*    37  *  ILLEGAL COMBINATION OF TYPES IN MULTIPLICATION *   E120   *
```

```
$131DUA4:[PROJECT.SYVAC.SYVAC.SC.P]DTBISECT.LIS;1                                                      18-MAR-

         *                                                              *          *
   43    *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A                      *          *
         *  CORRESPONDING BLOCK IF STATEMENT                            *   E324   *
         *                                                              *          *
   44    *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A                      *          *
         *  CORRESPONDING BLOCK IF STATEMENT                            *   E324   *
         *                                                              *          *
   45    *  ILLEGAL COMBINATION OF TYPES IN ADDITION OR                *          *
         *  SUBTRACTION                                                 *   E128   *
         *                                                              *          *
   49    *  ILLEGAL COMBINATION OF TYPES IN ADDITION OR                *          *
         *  SUBTRACTION                                                 *   E128   *
         *                                                              *          *
   51    *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A                      *          *
         *  CORRESPONDING BLOCK IF STATEMENT                            *   E324   *
         *                                                              *          *
   53    *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A                      *          *
         *  CORRESPONDING BLOCK IF STATEMENT                            *   E324   *
         *                                                              *          *
   54    *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A                      *          *
         *  CORRESPONDING BLOCK IF STATEMENT                            *   E324   *
         *                                                              *          *
   55    *  ELSEIF, ELSE OR ENDIF DOES NOT HAVE A                      *          *
         *  CORRESPONDING BLOCK IF STATEMENT                            *   E324   *
         *                                                              *          *
* ------ * ------------------------------------------------------------ * ------ *
         *                                                              *          *
  ....   *  SUBPROGRAM ALTERS DUMMY ARGUMENT ROOT                       *   W016   *
         *                                                              *          *
  ....   *  SUBPROGRAM ALTERS DUMMY ARGUMENT IOPS                       *   W016   *
         *                                                              *          *
* ------ * ------------------------------------------------------------ * ------ *
         *                                                              *          *
  ....   *  STANDARDS COMPLIANCE INDEX                                  *          *
         *  AVERAGE NUMBER OF ERRORS PER STATEMENT           0.397      *          *
         *  AVERAGE NUMBER OF WARNINGS PER STATEMENT         0.034      *   C004   *
         *                                                              *          *
**************************************************************************************
```

```
_$DISCUA4:[PROJECT.SYVAC.SYVAC.SC.PORTLIS]SCT.LIS;1                                       18-MAR-

****************************************************************
SC[illegible]TCOL
****************************************************************


****************************************************************
*                                                              *
*  F O R T R A N   P O R T A B I L I T Y   D O C U M E N T A T I O N  *
*                                                              *
****************************************************************
*        *                                          *          *
*  STAT  *                 MESSAGE                  *  KEY TO  *
*  NUMBER *                                         *  MANUAL  *
*        *                                          *          *
****************************************************************
*        *                                          *          *
*  ....  *  IDENTIFIER ABS MATCHES THE NAME OF AN INTRINSIC   *          *
*        *  FUNCTION                                *   P022   *
*        *                                          *          *
*  ....  *  IDENTIFIER INT MATCHES THE NAME OF AN INTRINSIC   *          *
*        *  FUNCTION                                *   P022   *
*        *                                          *          *
*  ....  *  NAME INT MATCHING NON-GENERIC INTRINSIC FUNCTION  *          *
*        *  NAME SHOULD HAVE EXPLICIT TYPING        *   P025   *
*        *                                          *          *
*  ....  *  INT IS SYSTEM DEPENDENT                 *   P027   *
*        *                                          *          *
*  ....  *  LOWER CASE ALPHA CHARACTERS FOUND IN THIS         *          *
*        *  PROGRAM UNIT                            *   P045   *
*        *                                          *          *
*  ----- *  --------------------------------------- *  ------  *
*        *                                          *          *
*  ....  *  PORTABILITY INDEX                       *          *
*        *  AVERAGE NUMBER OF PORTABILITY PROBLEMS  *          *
*        *  PER STATEMENT      0.096                *   C006   *
*        *                                          *          *
****************************************************************
```

```
_$13$DUA4:[PROJECT.SYVAC.SYVAC.SC.PEDTBISECT.LIS;1                                    18-M R-

*************************************************************************
SOFTOOL
*************************************************************************


*************************************************************************
*                                                                       *
*   F O R T R A N   A D D I T I O N A L   D O C U M E N T A T I O N     *
*                                                                       *
*************************************************************************
*        *                                                  *          *
*  STMT  *                    MESSAGE                       * KEY TO   *
* NUMBER *                                                  * MANUAL   *
*        *                                                  *          *
*************************************************************************
*        *                                                  *          *
*  ....  *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE     *          *
*        *  DEFINED BUT NOT REFERENCED:                     *   D021   *
*        *  J         F2        F1                          *          *
*        *                                                  *          *
*  ....  *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE     *          *
*        *  REFERENCED BUT NOT DEFINED BY THIS PROGRAM UNIT: *  D023   *
*        *  A         B         X3                          *          *
*        *                                                  *          *
*  ....  *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE     *          *
*        *  DEFINED BUT NOT REFERENCED BY THIS PROGRAM UNIT: *  D024   *
*        *  ROOT      IERR                                  *          *
*        *                                                  *          *
*  ....  *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE EACH *         *
*        *  DEFINED IN ONLY ONE STATEMENT:                  *   D025   *
*        *  J                                               *          *
*        *                                                  *          *
*  ....  *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE EACH *         *
*        *  DEFINED IN MORE THAN ONE STATEMENT:             *   D026   *
*        *  ROOT      IERR      X1        X2        F2        F1  *    *
*        *  F3                                              *          *
*        *                                                  *          *
*  ....  *  THE FOLLOWING IDENTIFIERS ARE DECLARED BY       *          *
*        *  DEFAULT:                                        *   D027   *
*        *  ZERO      ACCUR     ABS       A         B         FUNC *   *
*        *  F1        F2        F3        ROOT      X1        X2   *   *
*        *  INT       LOG       X3                          *          *
*        *                                                  *          *
*  ....  *  THE FOLLOWING IDENTIFIERS WERE ACTUAL ARGUMENTS *          *
*        *  IN A FUNCTION OR SUBROUTINE CALL:               *   D031   *
*        *  A         X1        X2        X3                *          *
*        *                                                  *          *
*  ....  *  THE FOLLOWING IDENTIFIERS WERE REFERENCED BUT   *          *
*        *  NOT DEFINED:                                    *   D051   *
*        *  NOTRY                                           *          *
*        *                                                  *          *
* ------ * ------------------------------------------------ * ------   *
*        *                                                  *          *
*  ....  *  DOCUMENTATION OPTION INDEX                      *          *
*        *  AVERAGE NUMBER OF DOCUMENTATION OPTION          *          *
*        *  MESSAGES PER STATEMENT    0.138                 *   C008   *
*        *                                                  *          *
```

```
_$1$DUA4:[PROJECT.SYVAC.SYVAC.SC.REITBISECT.LIS;1                                    18-MAR-

************************************************************
```

```
************************************************************************
SOFTOOL                THIRD PASS          MARCH 18, 1986    14:40
************************************************************************


  1          SUBROUTINE BISECT (A, B, ACCUR, FUNC, ROOT, IERR)
       C
       C********************************************************************** BISEC
       C  FINDS THE ROOT OF A FUNCTION BETWEEN TWO SPECIFIED LIMITS, USING
       C  A BISECTION ALGORITHM.  BASED ON THE DESCRIPTION BY R.W. HAMMING
       C  IN 'INTRODUCTION TO APPLIED NUMERICAL ANALYSIS', MCGRAW-HILL,
       C  1971, CHPT. 2.
       C
       C  86-FEB-25  VERSION 01  R.GOODWIN
       C  86-MAR-14  VERSION 02  R.GOODWIN
       C  86-MAR-18  VERSION 03  R.GOODWIN
       C
       C*************************************************************************
       C
  2          IMPLICIT LOGICAL (A-Z)
  3          DOUBLE PRECISION A
  4          DOUBLE PRECISION B
       C                        ! limits defining a range within which a
       C                          root is located [X]
  5          DOUBLE PRECISION ACCUR
       C                        ! relative accuracy desired in the root.  IERR is set
       C                          if ACCUR is less that 1.0D-12 [X/X]
  6          DOUBLE PRECISION FUNC
       C                        ! external function to be scanned.  Calls to FUNC
       C                          are of the form:  Z = FUNC (X) where X is the
       C                          independent variable.  [Z]
  7          DOUBLE PRECISION ROOT
       C                        ! contains the root found [X]
  8          INTEGER IERR
       C                        ! error code with the following values & meaning []:
       C                          IERR = 0  - no errors detected, root found
       C                          IERR = 1  - interval contains no roots or an even
       C                                      number of roots.  None have been found;
       C                                      re-use bisect with a smaller range.
       C                          IERR = 2  - A and B are identical and are not a
       C                                      root.
       C                          IERR = 3  - no roots found, but a pole has
       C                                      has probably been found.
       C                          IERR = 4  - specified accuracy is less that 1.0D-12
  9          INTEGER J
       C                        ! loop index []
 10          INTEGER NOTRY
       C                        ! number of bisections required to acheive the
       C                          specified accuracy. []
 11          DOUBLE PRECISION F1
 12          DOUBLE PRECISION F2
 13          DOUBLE PRECISION F3
       C                        ! temporary variables used to store values of FUNC
       C                          evaluated at X1, X2 and X3.  [Z]
 14          DOUBLE PRECISION X1
 15          DOUBLE PRECISION X2
 16          DOUBLE PRECISION X3
       C                        ! temporary variables used to store values of the
```

```
C                   independent variable required by FUNC.  [X]
      DOUBLE PRECISION ZERO
C                   ! set equal to 0.0000 for comparison; [X]
      DATA ZERO /0.0000/
C  ......CHECK FOR "REASONABLE" ACCURACY
      IF (ACCUR .LT. 1.0E-12) THEN
         IERR = 4
      ELSE IF ( ABS(A-B) .EQ. ZERO) THEN
C  ......ARE A AND B IDENTICAL AND AT A ROOT?
         IF (FUNC(A) .EQ. ZERO) THEN
            IERR = 0
            ROOT = A
         ELSE
            IERR = 3
         ENDIF
      ELSE
C  ......SEARCH FOR A ROOT BETWEEN A AND B INCLUSIVE.
         X1 = A
         X2 = B
         F2 = FUNC(X2)
         F1 = FUNC(X1)
C  .........A ROOT ONLY EXISTS IF FUNC(A) AND FUNC(B) HAVE OPPOSITE
C              SIGNS.  IF THEY HAVE THE SAME SIGN, THERE COULD BE ZERO
C              OR AN EVEN NUMBER OF ROOTS - AN ERROR CONDITION.
         IF (F1*F2 .GT. ZERO) THEN
            IERR = 1
C  .........IS THERE A ROOT AT EITHER EXTREME?
         ELSE IF (F1 .EQ. ZERO) THEN
            IERR = 0
            ROOT = X1
         ELSE IF (F2 .EQ. ZERO) THEN
            IERR = 0
            ROOT = X2
         ELSE
C  ...............EACH SEARCH WILL USE UP TO NTRY BISECTIONS, WHICH
C                 WILL GUARANTEE THE REQUIRED ACCURACY.  THE CONSTANT
C                 BELOW IS ALOG(2)
            NTRY = INT ( ( LOG( ABS(B-A) ) - LOG(ACCUR) )
     1                / 0.6931471806000 )
            DO 350 J = 1, NTRY
               X3 = 0.5000 * (X1 + X2)
               F3 = FUNC (X3)
               IF ( (F1 * F3) .GT. ZERO) THEN
C  ...............ROOT COULD BE BETWEEN X3 AND X2
                  X1 = X3
                  F1 = F3
               ELSE IF ( (F1 * F3) .LT. ZERO) THEN
C  ...............ROOT COULD BE BETWEEN X3 AND X1
                  X2 = X3
                  F2 = F3
               ELSE
C  ...............FOUND A ROOT BANG ON; SAVE IT
                  IERR = 0
                  ROOT = X3
                  GO TO 350
               ENDIF
  350       CONTINUE
C  ............SOMETHING HAS NOW BEEN LOCATED WITHING THE DESIRED
C              ACCURACY.  IT IS ASSUMED TO BE A ROOT IF F3 LIES
C              BETWEEN F1 AND F2; OTHERWISE IT IS (PROBABLY) A
```

```
C                   ROLE.  CALCULATE A NEW X3 AND F3 AND TEST.
            X3 = 0.5000 * (X1 + X2)
            F3 = FUNC(X3)
            IF ( (F3-F2)*(F1-F3) .GT. ZERO) THEN
               IERR = 0
               ROOT = X3
            ELSE
               IERR = 3
            ENDIF
         ENDIF
      ENDIF
  300 CONTINUE
C           WRITE (6,3333) IERR,X1,X2,X3,F1,F2,F3
C 3333      FORMAT (1H , 'BISECT IERR=',I3,/5X,'X1-X3,F1-F3:',/6E12.3)
      RETURN
      END
```

```
****************************************************************************
SOFTOOL
****************************************************************************


****************************************************************************
*                                                                          *
*    F O R T R A N   S T A N D A R D S   D O C U M E N T A T I O N        *
*                                                                          *
****************************************************************************
*        *                                                      *          *
* STMT   *                      MESSAGE                         * KEY TO   *
* NUMBER *                                                      * MANUAL   *
*        *                                                      *          *
****************************************************************************
*        *                                                      *          *
*    4.  *  ILLEGAL COMBINATION OF TYPES IN ADDITION OR         *          *
*        *   SUBTRACTION                                        *   E12c   *
*        *                                                      *          *
* ------ * ---------------------------------------------------- * ------   *
*        *                                                      *          *
*  ....  *  SUBPROGRAM ALTERS DUMMY ARGUMENT ERRT               *   W016   *
*        *                                                      *          *
*  ....  *  SUBPROGRAM ALTERS DUMMY ARGUMENT IERR               *   W01c   *
*        *                                                      *          *
* ------ * ---------------------------------------------------- * ------   *
*        *                                                      *          *
*  ....  *  STANDARDS COMPLIANCE INDEX                          *          *
*        *  AVERAGE NUMBER OF ERRORS PER STATEMENT       0.014  *          *
*        *  AVERAGE NUMBER OF WARNINGS PER STATEMENT     0.029  *   C004   *
*        *                                                      *          *
****************************************************************************
```

200

```
******************************************************************************
SOFTOOL
******************************************************************************


******************************************************************************
*                                                                            *
*   F O R T R A N   P O R T A B I L I T Y   D O C U M E N T A T I O N        *
*                                                                            *
******************************************************************************
*        *                                                       *          *
*  STMT  *                      MESSAGE                          *  KEY TO  *
* NUMBER *                                                       *  MANUAL  *
*        *                                                       *          *
******************************************************************************
*        *                                                       *          *
*     19 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     21 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     28 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     30 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     35 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     38 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     46 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     49 *  COMPARISON OF REAL OF DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*     60 *  COMPARISON OF REAL OR DOUBLE PRECISION IS SYSTEM     *          *
*        *  DEPENDENT                                            *   P015   *
*        *                                                       *          *
*   .... *  IDENTIFIER ABS MATCHES THE NAME OF AN INTRINSIC      *          *
*        *  FUNCTION                                             *   P022   *
*        *                                                       *          *
*   .... *  IDENTIFIER INT MATCHES THE NAME OF AN INTRINSIC      *          *
*        *  FUNCTION                                             *   P022   *
*        *                                                       *          *
*   .... *  NAME INT MATCHING NON-GENERIC INTRINSIC FUNCTION     *          *
*        *  NAME SHOULD HAVE EXPLICIT TYPING                     *   P025   *
*        *                                                       *          *
*   .... *  INT IS SYSTEM DEPENDENT                              *   P027   *
*        *                                                       *          *
*        *                                                       *          *
*   .... *  LOWER CASE ALPHA CHARACTERS FOUND IN THIS            *          *
*        *  PROGRAM UNIT                                         *   P045   *
*        *                                                       *          *
* ------ * ----------------------------------------------------- * ------   *
*        *                                                       *          *
*  ....  *  PORTABILITY INDEX                                  *         *
*        *  AVERAGE NUMBER OF PORTABILITY PROBLEMS             *         *
*        *  PER STATEMENT ---- 0.2                             *  C006   *
*        *                                                     *         *
**************************************************************************
```

```
***********************************************************************************
SOFTOOL
***********************************************************************************


***********************************************************************************
*                                                                                 *
*     F O R T R A N   A D D I T I O N A L   D O C U M E N T A T I O N             *
*                                                                                 *
***********************************************************************************
*          *                                                     *                *
*  STMT    *                    MESSAGE                          *                *
*  NUMBER  *                                                     * KEY TO         *
*          *                                                     * MANUAL         *
*          *                                                     *                *
***********************************************************************************
*          *                                                     *                *
*  ....    *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE        *                *
*          *  DEFINED BUT NOT REFERENCED:                        *   D021         *
*          *  J                                                  *                *
*          *                                                     *                *
*  ....    *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE        *                *
*          *  REFERENCED BUT NOT DEFINED BY THIS PROGRAM UNIT:   *   D023         *
*          *  A            ACCUR                                 *                *
*          *                                                     *                *
*  ....    *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE        *                *
*          *  DEFINED BUT NOT REFERENCED BY THIS PROGRAM UNIT:   *   D024         *
*          *  ROOT     ITER                                      *                *
*          *                                                     *                *
*  ....    *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE EACH   *                *
*          *  DEFINED IN ONLY ONE STATEMENT:                     *   D025         *
*          *  J        ITER                                      *                *
*          *                                                     *                *
*  ....    *  THE VALUES OF THE FOLLOWING IDENTIFIERS ARE EACH   *                *
*          *  DEFINED IN MORE THAN ONE STATEMENT:                *   D026         *
*          *  ROOT     ITER      F1        F2        F3       X1 *                *
*          *  X2       X3                                        *                *
*          *                                                     *                *
*  ....    *  THE FOLLOWING IDENTIFIERS ARE DECLARED BY          *                *
*          *  DEFAULT:                                           *   D027         *
*          *  ABS      INT       LOG                             *                *
*          *                                                     *                *
*  ....    *  THE FOLLOWING IDENTIFIERS WERE ACTUAL ARGUMENTS    *                *
*          *  IN A FUNCTION OR SUBROUTINE CALL:                  *   D031         *
*          *  A        ACCUR     X1        X2        X3       ABS*                *
*          *                                                     *                *
*  ....    *  THE FOLLOWING IDENTIFIERS WERE REFERENCED BUT      *                *
*          *  NOT DEFINED:                                       *   D051         *
*          *  NCTRY                                              *                *
*          *                                                     *                *
*          *                                                     *                *
*--------- * ----------------------------------------------------* -------        *
*          *                                                     *                *
*  ....    *  DOCUMENTATION OPTION INDEX                         *                *
*          *  AVERAGE NUMBER OF DOCUMENTATION OPTION             *                *
*          *  MESSAGES PER STATEMENT      0.114                  *   C008         *
*          *                                                     *                *
***********************************************************************************
```

APPENDIX F. SAMPLE RESULTS FROM STRUCTURED BASIS MODULE TESTING AND FUNCTIONAL TESTING

As part of the quality assurance procedures, structured basis module tests and functional tests were carried out for part of the code developed for the uranium tailings model used in SYVAC. Other testing procedures, applied to the bulk of the code developed, are described in Section 2.6.

Structured basis module and functional tests consisted of both the "black box" and "white box" approaches. In black box testing, the tester uses only the original minispecifications developed for the code. These specifications are used to establish a series of tests, using drivers, stubs and/or previously tested modules, that can be used to determine whether the code behaves as claimed. Input test data include extreme and illegal data as well as "normal" data. The objective of the black box approach is to use these tests and input data to determine whether or not the code performs according to the written specifications.

In the white box approach, the tester is given access to the code. New tests and input data are then used, as required, to ensure that any inherent logic not previously tested is performing correctly. White box testing is also used to check for potential programming errors, such as possible round-off problems or the use of non-standard code.

In both approaches, the testing is performed by a modeller or programmer who did not produce the original codes. The tester is challenged to find any errors or potential problems, and report on the status of the test results.

Structured basis module tests and functional tests were performed on the BISECT subroutine that was developed to locate the root of a user-supplied function. The specifications for BISECT, reproduced in Appendix B, were used in the black box testing, and a listing of the code was used in the white box testing.

The specifications for BISECT include several sample tests that were to be applied. The tester used these sample tests, as well as a number of additional tests that were designed to further test BISECT. A driver routine and several subroutines (used to define the user-supplied functions) were coded and run with BISECT.

Some examples of the white box tests are reproduced on the following pages. These tests (and others not given here) show that BISECT correctly meets its specifications. An inspection of a listing of BISECT was also carried out, as part of the white box approach. Several additional tests were attempted, and it was concluded that BISECT functioned as claimed.

```
  DAY
  20-MAR-1986 14:41:15
$ TYPE FUNCO2.FOR
        DOUBLE PRECISION FUNCTION FUNC (X)
C
C *** TEST FUNCTION FOR BISECT ***
C P. O'CONNOR 86-MAR-20
C
        DOUBLE PRECISION X
C
        FUNC = X - 1.1234
C
        RETURN
        END
$
$ LINK FUNCO2,DRBISECT,BISECT
Compilation warnings
        in module DRBISECT file WNRE$ESA:[OCONNOR.BISECT]DRBISECT.OBJ;2
$ D .EXE

Directory WNRE$ESA:[OCONNOR.BISECT]

DRBISECT.EXE;1             7/9        20-MAR-1986 14:38  (RWED,RWED,RE,)
FUNCO2.EXE;1               7/9        20-MAR-1986 14:42  (RWED,RWED,RE,)

Total of 2 files, 14/18 blocks.
$ RUN FUNCO2
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
-2, 2, .0001
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.123352050781250    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 9, .0000001
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.123399995267391    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 2, .0000001
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.123400032520294    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 2, 1E-9
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.123399999924004    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 2, 1E-10
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.123399999982212    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 2, 1E-11
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.123400000004040    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0,1\1\ 2, 1E-12
0, 2, 1E-12
 CALLING 'BISECT'                                ✓
 RETURN FROM 'BISECT' WITH ROOT =    1.12339999999492    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 2, 1E-13
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.12339999999492    AND ERROR CODE =
          4✓
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 2, 1E-14
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =    1.12339999999492    AND ERROR CODE =
          4✓
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY

*INTERRUPT*

$
```

```
 DAY
 20-MAR-1986 14:46:10
$ TYPE FUNC03.FOR
        DOUBLE PRECISION FUNCTION FUNC (X)
C
C *** TEST FUNCTION FOR BISECT ***
C P. O'CONNOR 86-MAR-20
C
        DOUBLE PRECISION X
C
        FUNC = COS(X)
C
        RETURN
        END
$
$ LINK FUNC03,DRBISECT,BISECT
Compilation warnings
        in module DRBISECT file WNRE$ESA:[OCONNOR.BISECT]DRBISECT.OBJ;2
$
$ RUN FUNC03
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 3, .000001
 CALLING 'BISECT'
                                           ✓
 RETURN FROM 'BISECT' WITH ROOT =    1.570796728134155    AND ERROR CODE =
           0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
3, 6, .000001
 CALLING 'BISECT'
                                                ✓
 RETURN FROM 'BISECT' WITH ROOT =    4.712388753890991    AND ERROR CODE =
           0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
6, 9, .000001
 CALLING 'BISECT'
                                                ✓
 RETURN FROM 'BISECT' WITH ROOT =    7.853982210159302    AND ERROR CODE =
           0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
2, 3, .01
 CALLING 'BISECT'   0 Roots
 RETURN FROM 'BISECT' WITH ROOT =    7.853982210159302    AND ERROR CODE =
           ① ✓
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 6, .01
 CALLING 'BISECT'  2 Roots
 RETURN FROM 'BISECT' WITH ROOT =    7.853982210159302    AND ERROR CODE =
           1 ✓
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 9, .0001
 CALLING 'BISECT'  3 Roots
                                              ✓
 RETURN FROM 'BISECT' WITH ROOT =    1.570838928222656    AND ERROR CODE =
           0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
-3, 0, .000001
 CALLING 'BISECT'      1 Root
 RETURN FROM 'BISECT' WITH ROOT =  -1.570796728134155    AND ERROR CODE =
           0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
```

```
-3, 6, .000001        3 Roots
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =  -1.570796191692352    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 8, .000001
 CALLING 'BISECT'        3 Roots
 RETURN FROM 'BISECT' WITH ROOT =   1.570796012878418    AND ERROR CODE =
          0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY

*INTERRUPT*

$
$
```

```
  DAY
  21-MAR-1986 14:19:22
$ TYPE FUNC04.FOR
        DOUBLE PRECISION FUNCTION FUNC (X)
C
C *** TEST FUNCTION FOR BISECT ***
C P. O'CONNOR 86-MAR-20
C
        DOUBLE PRECISION X
C
        FUNC = TAN(X)
C
        RETURN
        END
$
$ RUN FUNC04
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
1, 1, .0003
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =  0.0000000000000000E+00AND ERROR CODE =
            2
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
0, 0. 1E-9
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =  0.0000000000000000E+00AND ERROR CODE =
            0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
3.14159, 3.14159, .00001
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =  0.0000000000000000E+00AND ERROR CODE =
            2
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
3.14159, 3.14159, .01
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =  0.0000000000000000E+00AND ERROR CODE =
            2
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
6, 7, .1E-9
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =   6.283185307227541    AND ERROR CODE =
            0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
4, 3, 1E-9
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =   3.141592654399574    AND ERROR CODE =
            0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY
-4, -3, 1E-9
 CALLING 'BISECT'
 RETURN FROM 'BISECT' WITH ROOT =  -3.141592654399574    AND ERROR CODE =
            0
 ENTER THE LOW LIMIT, HIGH LIMIT, AND ACCURACY

*INTERRUPT*

$
```

APPENDIX G. DETAILED OUTPUT FROM SYVAC3

The following pages contain a listing of the input file (starting on page G-2) used in the SYVAC3 probabilistic simulation, and one of the SYVAC3 output files (starting on page G-12). The input file is read in a "free format" mode that is a part of the SYVAC3 executive modules. This format permits the use of lines of comment text, which are delimited by the exclamation mark (!) character. The output file contains a summary of the simulations, a list of the input data, a summary of the matching of the contents of the "include" files with the input variables, and a list of some statistics of the sampled and consequence variables.

```
! THIS IS FILE SV03.INP FOR THE SYVAC3-UTAP COMPARISON.  1986-MAY-7
!
! Descriptive case title:
 '500 RANDOM RUNS FOR SYVAC3-UTAP COMPARISON (FOR THE NUTP-WNRE CONTRACT)'
! Output type (may be 'LONG' or 'SHORT'):
'LONG'
! Optional outputs:
! 'PAR'    ! 'PAR' for all parameter values
! 'SUB'    ! 'SUB' for time series during submodel execution
! 'NDS'    ! 'NDS' for time series containing doses from individual nuclides
! 'CDS'    ! 'CDS' for time series containing doses combined from all nuclides
! Note: Use either SUB, NDS, or CDS to store trace output.
          !'PAR' 'SUB'
! Number of scenarios for this case:
       1  ! First run number
&    500  ! Requested number of "accepted" runs
&    500  ! Last run number (overrides requested number)
! Time series control:
    500       ! Cutoff time for simulation [years]
&   20  100  ! Minimum and maximum number of time steps
&   0.01      ! Target fractional error for time series representations
! Wasteform (may be 'U' for used fuel,
!                      'S' for solubility limited recycled fuel,
!                      'L' for leach rate limited recycled fuel):
'U'           ! If you don't need a wasteform, use U.
!
!
! List of INCLUDE files containing COMMON blocks of sampled, calculated, and
! consequence parameters:
   'SPAIR.INC'
   'SPDEP.INC'
   'SPLAKE.INC'
   'SPRATH.INC'
   'SPTIME.INC'
!
   'CPALL.INC'
!
   'CQFISH.INC'
   'CQMEAT.INC'
   'CQRND.INC'
   'CQVEG.INC'
   'CQWAT.INC'
   'CQXIN.INC'
   'CQTDOS.INC'
   'CQLAPH.INC'
'END'         ! End of INCLUDE list
!
! Parameter descriptions are grouped according to their sampling or calculation
! methods. There should be a separate method for each submodel (and seeds
! should not overlap). Valid methods are:
!     'RANDOM'   SEED   where SEED is an integer in (-2147483648,2147483647),
!     'QUANTILE' QFILE  where QFILE is the name of a file containing quantiles,
!     'ON FILE'  VFILE  where VFILE is the name of a file containing values,
!     'CALCULATED'      , and
!     'CONSEQUENCES'
! For RANDOM, QUANTILE, and ON FILE, parameter descriptions have the form
!     LONG_NAME  SHORT_NAME  SI_UNITS  SAVE_MARKER('S' or ' ')
!    &DISTRIBUTION_TYPE  DISTRIBUTION_PARAMETERS  BOUND_TYPE('Q' or 'V')  BOUNDS
!    &(if a piecewise uniform distribution) PIECEWISE_UNIFORM_DISTN_PARAMETERS
! DISTRIBUTION_PARAMETERS:
!     CONST:  constant value
!     NORML:  mu, sigma, bounds
!     LGNRM:  geometric mean, geometric standard deviation, bounds
!     UNIFM:  left end, right end, bounds
!     LGUFM:  left end, right end, bounds
!     TRIAN:  left end, mode, right end, bounds
!     BETA:   left end, right end, shape 1, shape 2, bounds
!     PCUFM:  number of classes, bounds,
!             <left end, right end, weight>
!     CORNR:  mu, sigma, correlation coefficient, bounds,
!             short name of independent correlated parameter
!     CORLN:  geometric mean, geometric standard deviation,
!             correlation coefficient, bounds, short name of
!             independent correlated parameter.
! For CALCULATED, parameter descriptions have the form
!     LONG_NAME  SHORT_NAME  SI_UNITS  SAVE_MARKER('S' or ' ')  LENGTH
! For CONSEQUENCES, parameter descriptions have the form
!     LONG_NAME  SHORT_NAME  SI_UNITS  SAVE_MARKER('S' or ' ')  LENGTH
!   &CONSEQUENCE_TYPE('LIT' or 'NUM')  ARTIFICIAL_VALUE
!
! If a subscripted variable appears in a COMMON block, then you must
!     declare a parameter (or consequence) for each value you want to use.
!     For example, given:
!        COMMON /ANYCOM/
!       1 XYZ(5)
!     you must declare XYZ(01), XYZ(02), ... XYZ(05) separately if you want
!     to use them (and yes, the 0's are needed!)
!
! Tailings parameters (all modules):
!
'RANDOM'   1759827131
!
   'ACTIVITY COEFFICIENT IN LAKE     '  'ACTCO        '  '-            '  'S'
&  'CONST'  0.8
   'EXPOSED TAILINGS SURFACE         '  'AEXPT        '  'm**2         '  'S'
&  'CONST'  5.00D05
   'EXPOSED TAILINGS SURFACE (II)    '  'ATAIL        '  'm**2         '  'S'
&  'CONST'  5.00D05
   'WATERSHED AREA                   '  'AWAT         '  'm**2         '  'S'
&  'CONST'  2.00D07
   'RATIO OF GAIN RATE TO LOSS RATE  '  'BETA         '  '-            '  'S'
&  'LGNRM'  0.18  1.35   'V'  0.1  0.328
   'WATER TO FISH TRANSFER COEFF.    '  'BF           '  'm**3/g       '  'S'
&  'LGNRM'  1.6D-05 3.5  'V'  1.0D-06 6.0D-04
   'INIT. SOLID PHASE ACT. OF RA-226'   'CONCRA       '  'Bq/g         '  'S'
&  'TRIAN'  5.0  11.0  15.0   'Q'  0.0   1.0
   'WHOLE BODY DCF FOR ING. RA-226   '  'DCFING       '  'Sv/Bq        '  'S'
&  'CONST'  3.20D-7
   'WHOLE BODY DCF FOR INHAL. RA-226'   'DCFINH       '  'Sv/Bq        '  'S'
```

```
&   'CONST'  2.6D-6
    'TIME STEP-SIZE                  '  'DELT          '  'a           ' 'S'
&   'CONST'  10.
    'DEPOSITION VELOCITY             '  'DEPVEL        '  'm/s         ' 'S'
&   'UNIFM'  0.003  0.03   'Q'   0.0   1.0
    'RA SOLID/AQ DIST. COEFFICIENT   '  'DISTCO        '  '-           ' 'S'
&   'UNIFM'  0.01  0.1   'Q'   0.0   1.0
    'DEPTH OF MIXING ZONE            '  'DMIX          '  'm           ' 'S'
&   'CONST'  1.00
    'DEPTH OF UNSAT. ZONE            '  'DUNSAT        '  'm           ' 'S'
&   'CONST'  1.00
    'DURATION OF EXPOSURE OF VEG. 1  '  'DUREXP(01)    '  's           ' 'S'
&   'UNIFM'  9.0D07  1.1D08   'Q'   0.0   1.0
    'DURATION OF EXPOSURE OF VEG. 2  '  'DUREXP(02)    '  's           ' 'S'
&   'UNIFM'  2.7D06  3.1D06   'Q'   0.0   1.0
    'DURATION OF EXPOSURE OF VEG. 3  '  'DUREXP(03)    '  's           ' 'S'
&   'UNIFM'  4.7D06  5.6D06   'Q'   0.0   1.0
    'DURATION OF EXPOSURE OF VEG. 4  '  'DUREXP(04)    '  's           ' 'S'
&   'UNIFM'  4.7D06  5.6D06   'Q'   0.0   1.0
    'EMANATION COEFFICIENT           '  'EMANAT        '  '-           ' 'S'
&   'TRIAN'  0.02  0.2  0.9   'Q'   0.0   1.0
    'EQUIL. FACTOR FOR RN+DAUGHTERS  '  'F             '  '-           ' 'S'
&   'TRIAN'  0.2  0.4  0.7   'Q'   0.0   1.0
    'ACTIVITY COEFF., DIVALENT IONS  '  'FACTCO        '  '-           ' 'S'
&   'TRIAN'  0.2  0.3  0.5   'Q'   0.0   1.0
    'FRAC. OF ANNUAL INTAKE OF VEG. 1'  'FAUT(01)      '  '-           ' 'S'
&   'TRIAN'  0.05  0.2  0.5   'Q'   0.0   1.0
    'FRAC. OF ANNUAL INTAKE OF VEG. 2'  'FAUT(02)      '  '-           ' 'S'
&   'TRIAN'  0.05  0.2  0.5   'Q'   0.0   1.0
    'FEED-MEAT TRANSF. COEFF., RA-226'  'FBI           '  'a/g         ' 'S'
&   'UNIFM'  0.3D-9  2.5D-9   'Q'   0.0   1.0
    'FRACTION ON EDIBLE PORTION, 1   '  'FDEPED(01)    '  '-           ' 'S'
&   'UNIFM'  0.8  1.0   'Q'   0.0   1.0
    'FRACTION ON EDIBLE PORTION, 2   '  'FDEPED(02)    '  '-           ' 'S'
&   'UNIFM'  0.8  1.0   'Q'   0.0   1.0
    'FRACTION ON EDIBLE PORTION, 3   '  'FDEPED(03)    '  '-           ' 'S'
&   'UNIFM'  0.8  1.0   'Q'   0.0   1.0
    'FRACTION ON EDIBLE PORTION, 4   '  'FDEPED(04)    '  '-           ' 'S'
&   'UNIFM'  0.05  0.15   'Q'   0.0   1.0
    'FRAC. OF DEPOSITION RETAINED, 1 '  'FDEPSF(01)    '  '-           ' 'S'
&   'UNIFM'  0.8  1.0   'Q'   0.0   1.0
    'FRAC. OF DEPOSITION RETAINED, 2 '  'FDEPSF(02)    '  '-           ' 'S'
&   'NORML'  0.47  0.3  'V'  0.01  1.0
    'FRAC. OF DEPOSITION RETAINED, 3 '  'FDEPSF(03)    '  '-           ' 'S'
&   'UNIFM'  0.1  0.3   'Q'   0.0   1.0
    'FRAC. OF DEPOSITION RETAINED, 4 '  'FDEPSF(04)    '  '-           ' 'S'
&   'UNIFM'  0.1  0.3   'Q'   0.0   1.0
    'FRAC., ANN. INTAKE OF LOC. FISH '  'FFISH         '  '-           ' 'S'
&   'CONST'  1.0
    'FEED INGESTION RATE             '  'FIR           '  'g/a         ' 'S'
&   'CONST'  2.5D03
    'AVERAGE ANNUAL INTAKE OF FISH   '  'FISHIN        '  'g/a         ' 'S'
&   'UNIFM'  4.0D03  12.0D03   'Q'   0.0   1.0
    'FRAC. OF VEG 3 GROWN LOCALLY    '  'FLOC(03)      '  '-           ' 'S'
&   'TRIAN'  0.1  0.3  0.6   'Q'   0.0   1.0
```

```
   'FRAC. OF VEG 4 GROWN LOCALLY     '  'FLOC(04)        '  '-              ' 'S'
&  'TRIAN'  0.1  0.3  0.6   'Q'   0.0   1.0
   'FRAC., ANN. INTAKE, LOC. CARIBOU'  'FMEAT              '-              ' 'S'
&  'UNIFM'  0.0  1.0   'Q'   0.0   1.0
   'ANNUAL DAYS WITH SIG. PRECIP.    '  'FPREC           '  'd              ' 'S'
&  'NORML'  123.0  0.8     'V'  121.0  125.0
   'MASS FRAC., GYPSUM IN TAILINGS   '  'FRACGP             '-              ' 'S'
&  'CONST'  0.05
   'FRAC., RA IN VEG. 3 AFTER PREP.  '  'FRACV(03)          '-              ' 'S'
&  'UNIFM'  0.5  1.0    'Q'   0.0   1.0
   'FRAC., RA IN VEG. 4 AFTER PREP.  '  'FRACV(04)       '  '-              ' 'S'
&  'UNIFM'  0.5  1.0    'Q'   0.0   1.0
   'FRAC. OF WATER INTAKE, LOCAL     '  'FWAT            '  '-              ' 'S'
&  'TRIAN'  0.4  0.6  0.9   'Q'   0.0   1.0
   '% OF TIME WITH HIGH WIND SPEEDS '  'FWIND1              '%              ' 'S'
&  'NORML'  35.0  0.3    'V'  34.1  35.9
   'FRAC., TIME WIND BLOWS TO RECEP.'  'FWIND2              '-              ' 'S'
&  'NORML'  0.1  8.0D-4    'V'  9.73D-2  0.103
   'HEIGHT OF SOURCE                 '  'HSRC               'm              ' 'S'
&  'UNIFM'   1.0  3.0   'Q'   0.0   1.0
   'RATIO OF KO2 TO [O2]             '  'KPRIME             '-              ' 'S'
&  'LGNRM'  5.0D-2  1.7     'V' 0.02  0.25
   'RA AQ-SOLID TRANSFER COEFFICIENT'  'KSUBDR           '  'm**3/g         ' 'S'
&  'LGNRM'  80.0D-6  3.5    'V' 1.9D-6  3.4E-3
   'SOLUBILITY EQM. CONST., GYPSUM   '  'KSUBSD             'mol**2/m**6    ' 'S'
&  'CONST'  25.0
   'CONV. FACTOR MOL TO BQ (RA-226)  '  'MOLTOB             'Bq/mol         ' 'S'
&  'CONST'  8.36D12
   '% MOISTURE CONTENT OF TAILINGS   '  'MPCH2O          '  '%              ' 'S'
&  'TRIAN'  5.0  15.0  25.0    'Q'   0.0   1.0
   'NET PRECIPITATION                '  'NPRCP           '  'm/a            ' 'S'
&  'NORML'  0.4  4.0D-3     'V'  0.39  0.41
   'O2 CONC. IN AIR                  '  'O2CONC             'mol/m**3       ' 'S'
&  'CONST'  9.04
   'O2 DIFFUSIVITY IN TAILINGS       '  'O2DIFF          '  'm**2/a         ' 'S'
&  'TRIAN'  30.0  70.0  130.0   'Q'   0.0   1.0
   'FRAC. OF YEAR AT LOCATION        '  'OF              '  '-              ' 'S'
&  'TRIAN'  0.4  0.6  0.9   'Q'   0.0   1.0
   'INIT. CONC., PYRITE IN TAILINGS '  'PYRITE          '  'mol/m**3       ' 'S'
&  'CONST'  730.
   'FRAC. OF TOTAL RA IN GYPSUM      '  'RAINGP          '  '-              ' 'S'
&  'TRIAN'   0.05  0.2  0.6   'Q'   0.0   1.0
   'FRAC., INGROWN RA THAT IS INSOL. '  'RAINTH             '-              ' 'S'
&  'TRIAN'   0.85  0.90  0.95   'Q'   0.0   1.0
   'RA-226 DECAY CONSTANT            '  'RALAM              '1/a            ' 'S'
&  'CONST'  4.33D-4
   'TAILINGS BULK DENSITY            '  'RHOTAL             'g/m**3         ' 'S'
&  'NORML'  1.4D6  0.1D6     'V' 1.1D6  1.7D6
   'RN-222 DECAY CONSTANT            '  'RNLAM              '1/s            ' 'S'
&  'CONST'  2.1D-6
   'SILT CONTENT OF TAILINGS SURFACE'  'SILT                '%              ' 'S'
&  'CONST'  30.0
   'SULPHATE PRODUCTION RATE         '  'SULPRD             'mol/g/a        ' 'S'
&  'LGNRM'  1.4D-6  1.5     'V'  0.4E-6  4.2D-6
   'RATE OF AMORPHOUS TH DISSOLUTION'  'THDISS              '1/a            ' 'S'
```

```
&  'UNIFM'  0.01  0.03   'Q'   0.0   1.0
   'FRAC. OF IMMOBILE TH IN TAILINGS'  'THFRAC        '  '-           ' 'S'
&  'TRIAN'  0.2  0.3  0.4   'Q'   0.0   1.0
   'TH-230 DECAY CONSTANT           '  'THLAM         '  '1/a         ' 'S'
&  'CONST'  8.66D-6
   'TIME STEP FOR CONSEQUENCES      '  'TSTEP         '  'a           ' 'S'
&  'CONST'  50.0
   'PORE VOLUME FRACTION            '  'TVOID         '  'm**3/m**3   ' 'S'
&  'TRIAN'  0.35  0.40  0.45   'Q'   0.0   1.0
   'INIT. MF OF UO2 IN ORE          '  'UO2TON        '  'kg/t        ' 'S'
&  'CONST'  1.2
   'MASS RATIO OF UO2 TO THO2       '  'UPERTH        '  '-           ' 'S'
&  'CONST'  3.0
   'MEAN WIND SPEED, SOURCE-RECEPTOR'  'UWIND         '  'm/s         ' 'S'
&  'NORML'  5.3  2.2D-2   'V' 5.2  5.4
   'VERTICAL DISPERSION COEFFICIENT '  'VDISP         '  'm           ' 'S'
&  'LGNRM'  60.0  1.08    'V' 47.6  75.7
   'AVERAGE ANNUAL INTAKE OF VEG. 3 '  'VEGIN(03)     '  'g/a         ' 'S'
&  'CONST'  1.57D5
   'AVERAGE ANNUAL INTAKE OF VEG. 4 '  'VEGIN(04)     '  'g/a         ' 'S'
&  'CONST'  7.00D4
   'WEATHERING LOSS DECAY RATE, 1   '  'VEGLAM(01)    '  '1/s         ' 'S'
&  'UNIFM'  2.0D-9  4.0D-9   'Q'   0.0   1.0
   'WEATHERING LOSS DECAY RATE, 2   '  'VEGLAM(02)    '  '1/s         ' 'S'
&  'UNIFM'  4.0D-7  6.0D-7   'Q'   0.0   1.0
   'WEATHERING LOSS DECAY RATE, 3   '  'VEGLAM(03)    '  '1/s         ' 'S'
&  'UNIFM'  4.0D-7  6.0D-7   'Q'   0.0   1.0
   'WEATHERING LOSS DECAY RATE, 4   '  'VEGLAM(04)    '  '1/s         ' 'S'
&  'UNIFM'  4.0D-7  6.0D-7   'Q'   0.0   1.0
   'LAKE VOLUME                     '  'VLAKE         '  'm**3        ' 'S'
&  'CONST'  1.6D7
   'AVERAGE ANNUAL INTAKE OF WATER  '  'WATIN         '  'm**3/a      ' 'S'
&  'CONST'  0.6
   'DISTANCE FROM SOURCE TO RECEPTOR'  'XDIST         '  'm           ' 'S'
&  'CONST'  3600.0
   'AVERAGE ANNUAL INHALATION RATE  '  'XINHIN        '  'm**3/a      ' 'S'
&  'CONST'  8400.0
   'RATE OF RA LOSS TO SEDIMENT     '  'XK1           '  '1/a         ' 'S'
&  'LGNRM'  0.43  1.85   'V'  0.126  1.47
   'RATE OF RA BURIAL TO DEEP SED.  '  'XK3           '  '1/a         ' 'S'
&  'LGNRM'  0.022  1.2   'V'  0.0153  0.0317
   'AVERAGE ANNUAL INTAKE OF CARIBOU'  'XMEATI        '  'g/a         ' 'S'
&  'CONST'  7.9D04
   'YIELD DENSITY, 1                '  'YDEN(01)      '  'g/m**2      ' 'S'
&  'UNIFM'  400.0  600.   'Q'   0.0   1.0
   'YIELD DENSITY, 2                '  'YDEN(02)      '  'g/m**2      ' 'S'
&  'UNIFM'  500.0 1000.   'Q'   0.0   1.0
   'YIELD DENSITY, 3                '  'YDEN(03)      '  'g/m**2      ' 'S'
&  'UNIFM'  1500. 2500.   'Q'   0.0   1.0
   'YIELD DENSITY, 4                '  'YDEN(04)      '  'g/m**2      ' 'S'
&  'UNIFM'  1500. 2500.   'Q'   0.0   1.0
'END'       ! End of TAILINGS sampled parameters
!
! Calculated parameters (actually calculated variables) used in
!    uranium tailings model.
```

```
!
'CALCULATED'
   'ACTCO**(1./4.)                    '  'ACT14      '  '-           '  'S' 1
   'ACTCO**(3./4.)                    '  'ACT34      '  '-           '  'S' 1
   'ACTCO**(9./4.)                    '  'ACT94      '  '-           '  'S' 1
   'SQ. ROOT OF EQBM. CONST.          '  'AEQBM      '  'mol/m**3    '  'S' 1
   'INIT. CONC. OF RA IN GYPSUM       '  'CRAGIN     '  'mol/m**3    '  'S' 1
   'INIT. CONC., SUL.,BTM OF UN.ZONE' 'CSINIT     '  'mol/m**3    '  'S' 1
   'INIT. SOLID PHASE CONC., TH-230 '  'CTH230     '  'mol/m**3    '  'S' 1
   'INIT. SOLID PHASE CONC., TH-232 '  'CTH232     '  'mol/m**3    '  'S' 1
   'RECIP., RN DIFFUSION PATH LENGTH' 'DIFLEN     '  '1/m         '  'S' 1
   'INIT. GYPSUM CONTENT IN TAILINGS' 'GYPBLK     '  'mol/m**3    '  'S' 1
   'RATE OF OXDN. OF PYRITE           '  'KRATE      '  '1/a         '  'S' 1
   'TOTAL INIT. CONC., RA IN SOLIDS '  'RADIUM     '  'mol/m**3    '  'S' 1
   'INIT. CONC., IMMOBILE RA IN TAIL' 'RASIMM     '  'mol/m**3    '  'S' 1
   'SULPH. FLUX FROM OXDN. OF PYRITE' 'SFLUX      '  'mol/m**2/a  '  'S' 1
   'TIME TO DEPLETE GYPSUM IN UN.ZON' 'TFINAL     '  'a           '  'S' 1
   'INIT. TOTAL TH IN SOLID PHASE    '  'THINIT     '  'mol/m**3    '  'S' 1
   'MF OF TH-230 TO TH-232            '  'THINTH     '  '-           '  'S' 1
   'RETENTION TIME IN LAKE            '  'XLAML       'a            '  'S' 1
!
'END'     ! End of calculated variables
!
'CONSEQUENCES'
!
   'DOSE FROM INGESTION OF FISH - 0 '  'DSFISH(01)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH -50 '  'DSFISH(02)     'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-100 '  'DSFISH(03)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-150 '  'DSFISH(04)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-200 '  'DSFISH(05)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-250 '  'DSFISH(06)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-300 '  'DSFISH(07)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-350 '  'DSFISH(08)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-400 '  'DSFISH(09)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-450 '  'DSFISH(10)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF FISH-500 '  'DSFISH(11)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'MAXIMUM VALUE OF DSFISH           '  'MAXFDS     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TIME OF MAXIMUM DSFISH            '  'TMXFDS     '  'a           '  'S' 1
&  'NUM'  '-1.0'
   'AVERAGE VALUE OF DSFISH           '  'AVGFDS     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INGESTION OF MEAT - 0 '  'DSMEAT(01)  '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
```

```
   'DOSE FROM INGESTION OF MEAT -50 '  'DSMEAT(02)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-100 '  'DSMEAT(03)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-150 '  'DSMEAT(04)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-200 '  'DSMEAT(05)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-250 '  'DSMEAT(06)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-300 '  'DSMEAT(07)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-350 '  'DSMEAT(08)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-400 '  'DSMEAT(09)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-450 '  'DSMEAT(10)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF MEAT-500 '  'DSMEAT(11)    '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'MAXIMUM VALUE OF DSMEAT         '  'MAXMDS        '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'TIME OF MAXIMUM DSMEAT          '  'TMXMDS        '  'a          ' 'S' 1
&  'NUM' '-1.0'
   'AVERAGE VALUE OF DSMEAT         '  'AVGMDS        '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN  - 0 '  'DSIRN(01)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN  -50 '  'DSIRN(02)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -100 '  'DSIRN(03)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -150 '  'DSIRN(04)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -200 '  'DSIRN(05)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -250 '  'DSIRN(06)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -300 '  'DSIRN(07)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -350 '  'DSIRN(08)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -400 '  'DSIRN(09)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -450 '  'DSIRN(10)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INHALATION OF RN -500 '  'DSIRN(11)     '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'MAXIMUM VALUE OF DSIRN          '  'MAXRND        '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'TIME OF MAXIMUM DSIRN           '  'TMXRND        '  'a          ' 'S' 1
&  'NUM' '-1.0'
   'AVERAGE VALUE OF DSIRN          '  'AVGRND        '  'Sv/a       ' 'S' 1
&  'NUM' '-1.0'
   'DOSE FROM INGESTION OF VEG. - 0 '  'DSIVEG(01)    '  'Sv/a       ' 'S' 1
```

```
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG. -50 '  'DSTVEG(02)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-100 '  'DSTVEG(03)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-150 '  'DSTVEG(04)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-200 '  'DSTVEG(05)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-250 '  'DSTVEG(06)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-300 '  'DSTVEG(07)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-350 '  'DSTVEG(08)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-400 '  'DSTVEG(09)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-450 '  'DSTVEG(10)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF VEG.-500 '  'DSTVEG(11)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'MAXIMUM VALUE OF DSTVEG         '  'MAXVDS          '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'TIME OF MAXIMUM DSTVEG          '  'TMXVDS          '  'a            '  'S' 1
&   'NUM' '-1.0'
    'AVERAGE VALUE OF DSTVEG         '  'AVGVDS          '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT. - 0 '  'DSWATR(01)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT. -50 '  'DSWATR(02)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-100 '  'DSWATR(03)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-150 '  'DSWATR(04)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-200 '  'DSWATR(05)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-250 '  'DSWATR(06)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-300 '  'DSWATR(07)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-350 '  'DSWATR(08)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-400 '  'DSWATR(09)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-450 '  'DSWATR(10)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'DOSE FROM INGESTION OF WAT.-500 '  'DSWATR(11)      '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'MAXIMUM VALUE OF DSWATR         '  'MAXWDS          '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
    'TIME OF MAXIMUM DSWATR          '  'TMXWDS          '  'a            '  'S' 1
&   'NUM' '-1.0'
    'AVERAGE VALUE OF DSWATR         '  'AVGWDS          '  'Sv/a         '  'S' 1
&   'NUM' '-1.0'
```

```
   'DOSE FROM INHALING DUST      - 0 '  'DSDUST(01)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST      -50 '  'DSDUST(02)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -100 '  'DSDUST(03)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -150 '  'DSDUST(04)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -200 '  'DSDUST(05)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -250 '  'DSDUST(06)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -300 '  'DSDUST(07)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -350 '  'DSDUST(08)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -400 '  'DSDUST(09)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -450 '  'DSDUST(10)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'DOSE FROM INHALING DUST     -500 '  'DSDUST(11)    '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'MAXIMUM VALUE OF DSDUST          '  'MAXXIN        '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TIME OF MAXIMUM DSDUST           '  'TMXXIN        '  'a           '  'S' 1
&  'NUM'  '-1.0'
   'AVERAGE VALUE OF DSDUST          '  'AVGXIN        '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE            - 0 '  'DSTOT(01)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE            -50 '  'DSTOT(02)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -100 '  'DSTOT(03)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -150 '  'DSTOT(04)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -200 '  'DSTOT(05)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -250 '  'DSTOT(06)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -300 '  'DSTOT(07)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -350 '  'DSTOT(08)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -400 '  'DSTOT(09)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -450 '  'DSTOT(10)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TOTAL ANNUAL DOSE           -500 '  'DSTOT(11)     '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'MAXIMUM VALUE OF DSTOT           '  'MAXTOT        '  'Sv/a        '  'S' 1
&  'NUM'  '-1.0'
   'TIME OF MAXIMUM DSTOT            '  'TMXTOT        '  'a           '  'S' 1
&  'NUM'  '-1.0'
   'AVERAGE VALUE OF DSTOT           '  'AVGTOT        '  'Sv/a        '  'S' 1
```

```
&  'NUM' '-1.0'
   'PH OF LAKE                 - 0 '  'LAKPH(01)    ' '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                 -50 '  'LAKPH(02)      '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -100 '  'LAKPH(03)      '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -150 '  'LAKPH(04)      '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -200 '  'LAKPH(05)      '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -250 '  'LAKPH(06)      '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -300 '  'LAKPH(07)    ' '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -350 '  'LAKPH(08)      '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -400 '  'LAKPH(09)      '-                ' 'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -450 '  'LAKPH(10)      '-                  'S' 1
&  'NUM' '-1.0'
   'PH OF LAKE                -500 '  'LAKPH(11)    ' '-                ' 'S' 1
&  'NUM' '-1.0'
   'MAXIMUM VALUE OF LAKPH         '  'MAXLPH          -                  'S' 1
&  'NUM  '-1.0'
   'TIME OF MAXIMUM LAKPH          '  'TMXLPH          a                ' 'S' 1
&  'NUM' '-1.0'
   'AVERAGE VALUE OF LAKPH         '   AVGLPH         '-                ' 'S' 1
&  'NUM' '-1.0'
!
'END        ! End of consequences
'END'       ! End of sampling methods
!
! Chains have the form WASTEFORM  NUCLIDE_ID NUCLIDE_ID ... NUCLIDE_ID:
! Chains are not compulsory, but if they are included, an inventory parameter
! is required for every nuclide, with the name and units:
!   'INVENTORY, V           NNNNN      ' 'INVTRY(01)       'mol/kg U    '
! where NNNNN is the name of the nuclide.
!
'END'            ! End of nuclide chains
```

```
SYVAC3 CASE: SV0303
BEGUN  14-MAY-86  16:21:27
TITLE: 500 RANDOM RUNS FOR SYVAC3-UTAP COMPARISON (FOR THE NUTP-WNRE CONTRACT)

  START AT RUN NUMBER ==================     1

  NUMBER OF ACCEPTED RUNS ==============   500

  LAST POSSIBLE RUN NUMBER =============   500

  OUTPUT TYPE ========================== LONG

  OPTIONAL FILES =======================

  UPPER TIME LIMIT ===================== 5.0000D+02

  MINIMUM NUMBER OF TIME STEPS ========= 20

  MAXIMUM NUMBER OF TIME STEPS ========= 100

  TARGET FRACTIONAL ERROR ============== 1.000000D-02

  WASTE FORM =========================== USED FUEL
1COMMON BLOCKS OF PARAMETERS IN INCLUDE FILES ARE:

 SPAIR.INC
 /SPAIR1/
 /SPAIR2/
 /SPAIR3/
 SPDEP.INC
 /SPDEP1/
 /SPDEP2/
 SPLAKE.INC
 /SPLAKE/
 SPRATH.INC
 /SPRATH/
 SPTIME.INC
 /SPTIME/
 CPALL.INC
 /CPALL /
 CQFISH.INC
 /CQFISH/
 CQMEAT.INC
 /CQMEAT/
 CQRND.INC
 /CQRND /
 CQVEG.INC
 /CQVEG /
 CQWAT.INC
 /CQWAT /
 CQXIN.INC
 /CQXIN /
 CQTDOS.INC
 /CQTDOS/
 CQLAPH.INC
```

```
/COLAPH/

1SAMPLING METHOD ============== RANDOM
 RANDOM SEED ================== 1759827131

 PARAMETERS SAMPLED BY RANDOM   ARE:


 ACTIVITY-COEFFICIENT IN LAKE       ACTCO         -            S    CONST
    8.000000D-01

 EXPOSED TAILINGS SURFACE           AEXPT         m**2         S    CONST
    5.000000D+05

 EXPOSED TAILINGS SURFACE (II)      ATAIL         m**2         S    CONST
    5.000000D+05

 WATERSHED AREA                     AWAT          m**2         S    CONST
    2.000000D+07

 RATIO OF GAIN RATE TO LOSS RATE    BETA          -            S    LGNRM
    1.800000D-01   1.350000D+00              1.000000D-01  3.280000D-01    0.02508    0.97722

 WATER TO FISH TRANSFER COEFF.      BF            m**3/g       S    LGNRM
    1.600000D-05   3.500000D+00              1.000000D-06  6.000000D-04    0.01344    0.99809

 INIT. SOLID PHASE ACT. OF RA-226   CONCRA        Bq/g         S    TRIAN
    5.000000E+00   1.100000E+01  1.500000E+01  5.000000D+00  1.500000D+01    0.00000    1.00000

 WHOLE BODY DCF FOR ING. RA-226     DCFING        Sv/Bq        S    CONST
    3.200000D-07

 WHOLE BODY DCF FOR INHAL. RA-226   DCFINH        Sv/Bq        S    CONST
    2.600000D-06

 TIME STEP-SIZE                     DELT          s            S    CONST
    1.000000D+01

 DEPOSITION VELOCITY                DEPVEL        m/s          S    UNIFM
    3.000000D-03   3.000000D-02              3.000000D-03  3.000000D-02    0.00000    1.00000

 RA SOLID/AQ DIST. COEFFICIENT      DISTCO        -            S    UNIFM
    1.000000D-02   1.000000D-01              1.000000D-02  1.000000D-01    0.00000    1.00000

 DEPTH OF MIXING ZONE               DMIX          m            S    CONST
    1.000000D+00

 DEPTH OF UNSAT. ZONE               DUNSAT        m            S    CONST
    1.000000D+00

 DURATION OF EXPOSURE OF VEG. 1     DUREXP(01)    s            S    UNIFM
    9.000000D+07   1.100000D+08              9.000000D+07  1.100000D+08    0.00000    1.00000

 DURATION OF EXPOSURE OF VEG. 2     DUREXP(02)    s            S    UNIFM
    2.700000D+06   3.100000D+06              2.700000D+06  3.100000D+06    0.00000    1.00000
```

```
DURATION OF EXPOSURE OF VEG. 3       DUREXP(03)       s               S    UNIFM
  4.700000D+06   5.600000D+06                              4.700000D+06   5.600000D+06    0.00000    1.00000

DURATION OF EXPOSURE OF VEG. 4       DUREXP(04)       s               S    UNIFM
  4.700000D+06   5.600000D+06                              4.700000D+06   5.600000D+06    0.00000    1.00000

EMANATION COEFFICIENT                EMANAT           -               S    TRIAN
  2.000000E-02   2.000000E-01   9.000000E-01               2.000000D-02   9.000000D-01    0.00000    1.00000

EQUIL. FACTOR FOR RN+DAUGHTERS       F                -               S    TRIAN
  2.000000E-01   4.000000E-01   7.000000E-01               2.000000D-01   7.000000D-01    0.00000    1.00000

ACTIVITY COEFF., DIVALENT IONS       FACTCO           -               S    TRIAN
  2.000000E-01   3.000000E-01   5.000000E-01               2.000000D-01   5.000000D-01    0.00000    1.00000

FRAC. OF ANNUAL INTAKE OF VEG. 1     FAUT(01)         -               S    TRIAN
  5.000000E-02   2.000000E-01   5.000000E-01               5.000000D-02   5.000000D-01    0.00000    1.00000

FRAC. OF ANNUAL INTAKE OF VEG. 2     FAUT(02)         -               S    TRIAN
  5.000000E-02   2.000000E-01   5.000000E-01               5.000000D-02   5.000000D-01    0.00000    1.00000

FEED-MEAT TRANSF. COEFF., RA-226     FBI              a/g             S    UNIFM
  3.000000D-10   2.500000D-09                              3.000000D-10   2.500000D-09    0.00000    1.00000

FRACTION ON EDIBLE PORTION, 1        FDEPED(01)       -               S    UNIFM
  8.000000D-01   1.000000D+00                              8.000000D-01   1.000000D+00    0.00000    1.00000

FRACTION ON EDIBLE PORTION, 2        FDEPED(02)       -               S    UNIFM
  8.000000D-01   1.000000D+00                              8.000000D-01   1.000000D+00    0.00000    1.00000

FRACTION ON EDIBLE PORTION, 3        FDEPED(03)       -               S    UNIFM
  8.000000D-01   1.000000D+00                              8.000000D-01   1.000000D+00    0.00000    1.00000

FRACTION ON EDIBLE PORTION, 4        FDEPED(04)       -               S    UNIFM
  5.000000D-02   1.500000D-01                              5.000000D-02   1.500000D-01    0.00000    1.00000

FRAC. OF DEPOSITION RETAINED, 1      FDEPSF(01)       -               S    UNIFM
  8.000000D-01   1.000000D+00                              8.000000D-01   1.000000D+00    0.00000    1.00000

FRAC. OF DEPOSITION RETAINED, 2      FDEPSF(02)       -               S    NORML
  4.700000D-01   3.000000D-01                              1.000000D-02   1.000000D+00    0.06260    0.96136

FRAC. OF DEPOSITION RETAINED, 3      FDEPSF(03)       -               S    UNIFM
  1.000000D-01   3.000000D-01                              1.000000D-01   3.000000D-01    0.00000    1.00000

FRAC. OF DEPOSITION RETAINED, 4      FDEPSF(04)       -               S    UNIFM
  1.000000D-01   3.000000D-01                              1.000000D-01   3.000000D-01    0.00000    1.00000

FRAC., ANN. INTAKE OF LOC. FISH      FFISH            -               S    CONST
  1.000000D+00

FEED INGESTION RATE                  FIR              g/a             S    CONST
  2.500000D+03
```

```
AVERAGE ANNUAL INTAKE OF FISH        FISHIN             g/a              S     UNIFM
   4.000000D+03   1.200000D+04                                 4.000000D+03   1.200000D+04    0.00000    1.00000

FRAC. OF VEG 3 GROWN LOCALLY         FLOC(03)           -                S     TRIAN
   1.000000E-01   3.000000E-01   6.000000E-01                  1.000000D-01   6.000000D-01    0.00000    1.00000

FRAC. OF VEG 4 GROWN LOCALLY         FLOC(04)           -                S     TRIAN
   1.000000E-01   3.000000E-01   6.000000E-01                  1.000000D-01   6.000000D-01    0.00000    1.00000

FRAC., ANN. INTAKE, LOC. CARIBOU     FMEAT              -                S     UNIFM
   0.000000D-01   1.000000D+00                                 0.000000D-01   1.000000D+00    0.00000    1.00000

ANNUAL DAYS WITH SIG. PRECIP.        FPREC              d                S     NORML
   1.230000D+02   8.000000D-01                                 1.210000D+02   1.250000D+02    0.00621    0.99379

MASS FRAC., GYPSUM IN TAILINGS       FRACGP             -                S     CONST
   5.000000D-02

FRAC., RA IN VEG. 3 AFTER PREP.      FRACV(03)          -                S     UNIFM
   5.000000D-01   1.000000D+00                                 5.000000D-01   1.000000D+00    0.00000    1.00000

FRAC., RA IN VEG. 4 AFTER PREP.      FRACV(04)          -                S     UNIFM
   5.000000D-01   1.000000D+00                                 5.000000D-01   1.000000D+00    0.00000    1.00000

FRAC. OF WATER INTAKE, LOCAL         FWAT               -                S     TRIAN
   4.000000E-01   6.000000E-01   9.000000E-01                  4.000000D-01   9.000000D-01    0.00000    1.00000

% OF TIME WITH HIGH WIND SPEEDS      FWIND1             %                S     NORML
   3.500000D+01   3.000000D-01                                 3.410000D+01   3.590000D+01    0.00135    0.99865

FRAC., TIME WIND BLOWS TO RECEP.     FWIND2             -                S     NORML
   1.000000D-01   8.000000D-04                                 9.730000D-02   1.030000D-01    0.00037    0.99991

HEIGHT OF SOURCE                     HSRC               m                S     UNIFM
   1.000000D+00   3.000000D+00                                 1.000000D+00   3.000000D+00    0.00000    1.00000

RATIO OF KO2 TO [O2]                 KPRIME             -                S     LGNRM
   5.000000D-02   1.700000D+00                                 2.000000D-02   2.500000D-01    0.04210    0.99879

RA AQ-SOLID TRANSFER COEFFICIENT     KSUBDR             m**3/g           S     LGNRM
   8.000000D-05   3.500000D+00                                 1.900000D-06   3.400000D-03    0.00142    0.99862

SOLUBILITY EQM. CONST., GYPSUM       KSUBSO             mol**2/m**6      S     CONST
   2.500000D+01

CONV. FACTOR MOL TO BQ (RA-226)      MOLTOB             Bq/mol           S     CONST
   8.360000D+12

% MOISTURE CONTENT OF TAILINGS       MPCH2O             %                S     TRIAN
   5.000000E+00   1.500000E+01   2.500000E+01                  5.000000D+00   2.500000D+01    0.00000    1.00000

NET PRECIPITATION                    NPRCP              m/a              S     NORML
   4.000000D-01   4.000000D-03                                 3.900000D-01   4.100000D-01    0.00621    0.99379

O2 CONC. IN AIR                      O2CONC             mol/m**3         S     CONST
```

```
   9.040000D+00

02 DIFFUSIVITY IN TAILINGS          02DIFF              m**2/a          S     TRIAN
   3.000000E+01   7.000000E+01   1.300000E+02              3.000000D+01  1.300000D+02    0.00000     1.00000

FRAC. OF YEAR AT LOCATION           OF                  -               S     TRIAN
   4.000000E-01   6.000000E-01   9.000000E-01              4.000000D-01  9.000000D-01    0.00000     1.00000

INIT. CONC., PYRITE IN TAILINGS     PYRITE              mol/m**3        S     CONST
   7.300000D+02

FRAC. OF TOTAL RA IN GYPSUM         RAINGP              -               S     TRIAN
   5.000000E-02   2.000000E-01   6.000000E-01              5.000000D-02  6.000000D-01    0.00000     1.00000

FRAC., INGROWN RA THAT IS INSOL.    RAINTH              -               S     TRIAN
   8.500000E-01   9.000000E-01   9.500000E-01              8.500000D-01  9.500000D-01    0.00000     1.00000

RA-226 DECAY CONSTANT               RALAM               1/a             S     CONST
   4.330000D-04

TAILINGS BULK DENSITY               RHOTAL              g/m**3          S     NORML
   1.400000D+06   1.000000D+05                             1.100000D+06  1.700000D+06    0.00135     0.99865

RN-222 DECAY CONSTANT               RNLAM               1/s             S     CONST
   2.100000D-06

SILT CONTENT OF TAILINGS SURFACE    SILT                %               S     CONST
   3.000000D+01

SULPHATE PRODUCTION RATE            SULPRD              mol/g/a         S     LGNRM
   1.400000D-06   1.500000D+00                             4.000000D-07  4.200000D-06    0.00100     0.99663

RATE OF AMORPHOUS TH DISSOLUTION    THDISS              1/a             S     UNIFM
   1.000000D-02   3.000000D-02                             1.000000D-02  3.000000D-02    0.00000     1.00000

FRAC. OF IMMOBILE TH IN TAILINGS    THFRAC              -               S     TRIAN
   2.000000E-01   3.000000E-01   4.000000E-01              2.000000D-01  4.000000D-01    0.00000     1.00000

TH-230 DECAY CONSTANT               THLAM               1/a             S     CONST
   8.660000D-06

TIME STEP FOR CONSEQUENCES          TSTEP               a               S     CONST
   5.000000D+01

PORE VOLUME FRACTION                TVOID               m**3/m**3       S     TRIAN
   3.500000E-01   4.000000E-01   4.500000E-01              3.500000D-01  4.500000D-01    0.00000     1.00000

INIT. MF OF UO2 IN ORE              UO2TON              kg/t            S     CONST
   1.200000D+00

MASS RATIO OF UO2 TO THO2           UPERTH              -               S     CONST
   3.000000D+00

MEAN WIND SPEED, SOURCE-RECEPTOR    UWIND               m/s             S     NORML
   5.300000D+00   2.200000D-02                             5.200000D+00  5.400000D+00    0.00000     1.00000
```

```
VERTICAL DISPERSION COEFFICIENT      VDISP          m                 S     LGNRM
   6.000000D+01   1.080000D+00                             4.760000D+01  7.570000D+01    0.00131     0.99874

AVERAGE ANNUAL INTAKE OF VEG. 3      VEGIN(03)      g/a               S     CONST
   1.570000D+05

AVERAGE ANNUAL INTAKE OF VEG. 4      VEGIN(04)      g/a               S     CONST
   7.000000D+04

WEATHERING LOSS DECAY RATE, 1        VEGLAM(01)     1/s               S     UNIFM
   2.000000D-09   4.000000D-09                             2.000000D-09  4.000000D-09    0.00000     1.00000

WEATHERING LOSS DECAY RATE, 2        VEGLAM(02)     1/s               S     UNIFM
   4.000000D-07   6.000000D-07                             4.000000D-07  6.000000D-07    0.00000     1.00000

WEATHERING LOSS DECAY RATE, 3        VEGLAM(03)     1/s               S     UNIFM
   4.000000D-07   6.000000D-07                             4.000000D-07  6.000000D-07    0.00000     1.00000

WEATHERING LOSS DECAY RATE, 4        VEGLAM(04)     1/s               S     UNIFM
   4.000000D-07   6.000000D-07                             4.000000D-07  6.000000D-07    0.00000     1.00000

LAKE VOLUME                          VLAKE          m**3              S     CONST
   1.600000D+07

AVERAGE ANNUAL INTAKE OF WATER       WATIN          m**3/a            S     CONST
   6.000000D-01

DISTANCE FROM SOURCE TO RECEPTOR     XDIST          m                 S     CONST
   3.600000D+03

AVERAGE ANNUAL INHALATION RATE       XINHIN         m**3/a            S     CONST
   8.400000D+03

RATE OF RA LOSS TO SEDIMENT          XK1            1/a               S     LGNRM
   4.300000D-01   1.850000D+00                             1.260000D-01  1.470000D+00    0.02300     0.97715

RATE OF RA BURIAL TO DEEP SED.       XK3            1/a               S     LGNRM
   2.200000D-02   1.200000D+00                             1.530000D-02  3.170000D-02    0.02318     0.97744

AVERAGE ANNUAL INTAKE OF CARIBOU     XMEATI         g/a               S     CONST
   7.900000D+04

YIELD DENSITY, 1                     YDEN(01)       g/m**2            S     UNIFM
   4.000000D+02   6.000000D+02                             4.000000D+02  6.000000D+02    0.00000     1.00000

YIELD DENSITY, 2                     YDEN(02)       g/m**2            S     UNIFM
   5.000000D+02   1.000000D+03                             5.000000D+02  1.000000D+03    0.00000     1.00000

YIELD DENSITY, 3                     YDEN(03)       g/m**2            S     UNIFM
   1.500000D+03   2.500000D+03                             1.500000D+03  2.500000D+03    0.00000     1.00000

YIELD DENSITY, 4                     YDEN(04)       g/m**2            S     UNIFM
   1.500000D+03   2.500000D+03                             1.500000D+03  2.500000D+03    0.00000     1.00000
```

```
1SAMPLING METHOD ============== CALCULATED

 CALCULATED PARAMETERS ARE:

 ACTCO**(1./4.)                     ACT14             -              S      1
 ACTCO**(3./4.)                     ACT34             -              S      1
 ACTCO**(9./4.)                     ACT94             -              S      1
 SQ. ROOT OF EQBM. CONST.           AEQUM             mol/m**3       S      1
 INIT. CONC. OF RA IN GYPSUM        CRAGIN            mol/m**3       S      1
 INIT. CONC., SUL.,BTM OF UN.ZONE   CSINIT            mol/m**3       S      1
 INIT. SOLID PHASE CONC., TH-230    CTH230            mol/m**3       S      1
 INIT. SOLID PHASE CONC., TH-232    CTH232            mol/m**3       S      1
 RECIP., RN DIFFUSION PATH LENGTH   DIFLEN            1/m            S      1
 INIT. GYPSUM CONTENT IN TAILINGS   GYPBLK            mol/m**3       S      1
 RATE OF OXDN. OF PYRITE            KRATE             1/a            S      1
 TOTAL INIT. CONC., RA IN SOLIDS    RADIUM            mol/m**3       S      1
 INIT. CONC., IMMOBILE RA IN TAIL   RASIMM            mol/m**3       S      1
 SULPH. FLUX FROM OXDN. OF PYRITE   SFLUX             mol/m**2/a     S      1
 TIME TO DEPLETE GYPSUM IN UN.ZON   TFINAL            a              S      1
 INIT. TOTAL TH IN SOLID PHASE      THINIT            mol/m**3       S      1
 MF OF TH-230 TO TH-232             THINTH            -              S      1
 RETENTION TIME IN LAKE             XLAML             a              S      1

1SAMPLING METHOD ============== CONSEQUENCES

 CONSEQUENCES ARE:

 DOSE FROM INGESTION OF FISH - 0    DSFISH(01)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH -50    DSFISH(02)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-100    DSFISH(03)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-150    DSFISH(04)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-200    DSFISH(05)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-250    DSFISH(06)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-300    DSFISH(07)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-350    DSFISH(08)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-400    DSFISH(09)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-450    DSFISH(10)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF FISH-500    DSFISH(11)        Sv/a           S      1     NUM      -1.0
 MAXIMUM VALUE OF DSFISH            MAXFDS            Sv/a           S      1     NUM      -1.0
 TIME OF MAXIMUM DSFISH             TMXFDS            a              S      1     NUM      -1.0
 AVERAGE VALUE OF DSFISH            AVGFDS            Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT - 0    DSMEAT(01)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT -50    DSMEAT(02)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-100    DSMEAT(03)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-150    DSMEAT(04)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-200    DSMEAT(05)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-250    DSMEAT(06)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-300    DSMEAT(07)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-350    DSMEAT(08)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-400    DSMEAT(09)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-450    DSMEAT(10)        Sv/a           S      1     NUM      -1.0
 DOSE FROM INGESTION OF MEAT-500    DSMEAT(11)        Sv/a           S      1     NUM      -1.0
 MAXIMUM VALUE OF DSMEAT            MAXMDS            Sv/a           S      1     NUM      -1.0
 TIME OF MAXIMUM DSMEAT             TMXMDS            a              S      1     NUM      -1.0
 AVERAGE VALUE OF DSMEAT            AVGMDS            Sv/a           S      1     NUM      -1.0
```

DOSE FROM INHALATION OF RN - 0	DSIRN(01)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -50	DSIRN(02)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -100	DSIRN(03)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -150	DSIRN(04)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -200	DSIRN(05)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -250	DSIRN(06)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -300	DSIRN(07)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -350	DSIRN(08)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -400	DSIRN(09)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -450	DSIRN(10)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALATION OF RN -500	DSIRN(11)	Sv/a	S	1	NUM	-1.0
MAXIMUM VALUE OF DSIRN	MAXRND	Sv/a	S	1	NUM	-1.0
TIME OF MAXIMUM DSIRN	TMXRND	a	S	1	NUM	-1.0
AVERAGE VALUE OF DSIRN	AVGRND	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG. - 0	DSTVEG(01)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG. -50	DSTVEG(02)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-100	DSTVEG(03)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-150	DSTVEG(04)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-200	DSTVEG(05)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-250	DSTVEG(06)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-300	DSTVEG(07)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-350	DSTVEG(08)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-400	DSTVEG(09)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-450	DSTVEG(10)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF VEG.-500	DSTVEG(11)	Sv/a	S	1	NUM	-1.0
MAXIMUM VALUE OF DSTVEG	MAXVDS	Sv/a	S	1	NUM	-1.0
TIME OF MAXIMUM DSTVEG	TMXVDS	a	S	1	NUM	-1.0
AVERAGE VALUE OF DSTVEG	AVGVDS	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT. - 0	DSWATR(01)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT. -50	DSWATR(02)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-100	DSWATR(03)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-150	DSWATR(04)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-200	DSWATR(05)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-250	DSWATR(06)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-300	DSWATR(07)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-350	DSWATR(08)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-400	DSWATR(09)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-450	DSWATR(10)	Sv/a	S	1	NUM	-1.0
DOSE FROM INGESTION OF WAT.-500	DSWATR(11)	Sv/a	S	1	NUM	-1.0
MAXIMUM VALUE OF DSWATR	MAXWDS	Sv/a	S	1	NUM	-1.0
TIME OF MAXIMUM DSWATR	TMXWDS	a	S	1	NUM	-1.0
AVERAGE VALUE OF DSWATR	AVGWDS	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST - 0	DSDUST(01)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -50	DSDUST(02)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -100	DSDUST(03)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -150	DSDUST(04)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -200	DSDUST(05)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -250	DSDUST(06)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -300	DSDUST(07)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -350	DSDUST(08)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -400	DSDUST(09)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -450	DSDUST(10)	Sv/a	S	1	NUM	-1.0
DOSE FROM INHALING DUST -500	DSDUST(11)	Sv/a	S	1	NUM	-1.0
MAXIMUM VALUE OF DSDUST	MAXXIN	Sv/a	S	1	NUM	-1.0
TIME OF MAXIMUM DSDUST	TMAXIN	a	S	1	NUM	-1.0

```
AVERAGE VALUE OF DSDUST            AVGXIN       Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          - 0     DSTOT(01)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -50     DSTOT(02)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -100    DSTOT(03)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -150    DSTOT(04)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -200    DSTOT(05)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -250    DSTOT(06)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -300    DSTOT(07)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -350    DSTOT(08)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -400    DSTOT(09)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -450    DSTOT(10)    Sv/a       S    1    NUM    -1.0
TOTAL ANNUAL DOSE          -500    DSTOT(11)    Sv/a       S    1    NUM    -1.0
MAXIMUM VALUE OF DSTOT             MAXTOT       Sv/a       S    1    NUM    -1.0
TIME OF MAXIMUM DSTOT              TMXTOT       a          S    1    NUM    -1.0
AVERAGE VALUE OF DSTOT             AVGTOT       Sv/a       S    1    NUM    -1.0
PH OF LAKE                 - 0     LAKPH(01)    -          S    1    NUM    -1.0
PH OF LAKE                 -50     LAKPH(02)    -          S    1    NUM    -1.0
PH OF LAKE                 -100    LAKPH(03)    -          S    1    NUM    -1.0
PH OF LAKE                 -150    LAKPH(04)    -          S    1    NUM    -1.0
PH OF LAKE                 -200    LAKPH(05)    -          S    1    NUM    -1.0
PH OF LAKE                 -250    LAKPH(06)    -          S    1    NUM    -1.0
PH OF LAKE                 -300    LAKPH(07)    -          S    1    NUM    -1.0
PH OF LAKE                 -350    LAKPH(08)    -          S    1    NUM    -1.0
PH OF LAKE                 -400    LAKPH(09)    -          S    1    NUM    -1.0
PH OF LAKE                 -450    LAKPH(10)    -          S    1    NUM    -1.0
PH OF LAKE                 -500    LAKPH(11)    -          S    1    NUM    -1.0
MAXIMUM VALUE OF LAKPH             MAXLPH       -          S    1    NUM    -1.0
TIME OF MAXIMUM LAKPH              TMXLPH       a          S    1    NUM    -1.0
AVERAGE VALUE OF LAKPH             AVGLPH       -          S    1    NUM    -1.0
** CHECKI ** DCFINH        /  SPAIR1   :   T
** CHECKI ** DEPVEL        /  SPAIR1   :   T
** CHECKI ** DUREXP(01)    /  SPAIR1   :   T
** CHECKI ** DUREXP(02)    /  SPAIR1   :   T
** CHECKI ** DUREXP(03)    /  SPAIR1   :   T
** CHECKI ** DUREXP(04)    /  SPAIR1   :   T
** CHECKI ** F             /  SPAIR1   :   T
** CHECKI ** FAUT(01)      /  SPAIR1   :   T
** CHECKI ** FAUT(02)      /  SPAIR1   :   T
** CHECKI ** FBI           /  SPAIR1   :   T
** CHECKI ** FDEPED(01)    /  SPAIR1   :   T
** CHECKI ** FDEPED(02)    /  SPAIR1   :   T
** CHECKI ** FDEPED(03)    /  SPAIR1   :   T
** CHECKI ** FDEPED(04)    /  SPAIR1   :   T
** CHECKI ** FDEPSF(01)    /  SPAIR1   :   T
** CHECKI ** FDEPSF(02)    /  SPAIR1   :   T
** CHECKI ** FDEPSF(03)    /  SPAIR1   :   T
** CHECKI ** FDEPSF(04)    /  SPAIR1   :   T
** CHECKI ** FIR           /  SPAIR1   :   T
** CHECKI ** FLOC(01)      /  SPAIR2   :   F

**(CHECKI) PARAMETER FLOC(01)       IS IN COMMON BLOCK SPAIR2 BUT NOT IN THE INPUT FILE.
** CHECKI ** FLOC(02)      /  SPAIR2   :   F

**(CHECKI) PARAMETER FLOC(02)       IS IN COMMON BLOCK SPAIR2 BUT NOT IN THE INPUT FILE.
** CHECKI ** FLOC(03)      /  SPAIR2   :   T
** CHECKI ** FLOC(04)      /  SPAIR2   :    T
** CHECKI ** FMEAT         /  SPAIR2   :    T
** CHECKI ** FRACV(01)     /  SPAIR2   :    F

**(CHECKI) PARAMETER FRACV(01)      IS IN COMMON BLOCK SPAIR2 BUT NOT IN THE INPUT FILE.
** CHECKI ** FRACV(02)     /  SPAIR2   :    F

**(CHECKI) PARAMETER FRACV(02)      IS IN COMMON BLOCK SPAIR2 BUT NOT IN THE INPUT FILE.
** CHECKI ** FRACV(03)     /  SPAIR2   :    T
** CHECKI ** FRACV(04)     /  SPAIR2   :    T
** CHECKI ** FWIND2        /  SPAIR2   :    T
** CHECKI ** HSRC          /  SPAIR2   :    T
** CHECKI ** OF            /  SPAIR2   :    T
** CHECKI ** UWIND         /  SPAIR2   :    T
** CHECKI ** VDISP         /  SPAIR2   :    T
** CHECKI ** VEGIN(01)     /  SPAIR2   :    F

**(CHECKI) PARAMETER VEGIN(01)      IS IN COMMON BLOCK SPAIR2 BUT NOT IN THE INPUT FILE.
** CHECKI ** VEGIN(02)     /  SPAIR2   :    F

**(CHECKI) PARAMETER VEGIN(02)      IS IN COMMON BLOCK SPAIR2 BUT NOT IN THE INPUT FILE.
** CHECKI ** VEGIN(03)     /  SPAIR2   :    T
** CHECKI ** VEGIN(04)     /  SPAIR2   :    T
** CHECKI ** VEGLAM(01)    /  SPAIR3   :    T
** CHECKI ** VEGLAM(02)    /  SPAIR3   :    T
** CHECKI ** VEGLAM(03)    /  SPAIR3   :    T
** CHECKI ** VEGLAM(04)    /  SPAIR3   :    T
** CHECKI ** XDIST         /  SPAIR3   :    T
** CHECKI ** XINHIN        /  SPAIR3   :    T
** CHECKI ** XMEATI        /  SPAIR3   :    T
** CHECKI ** YDEN(01)      /  SPAIR3   :    T
** CHECKI ** YDEN(02)      /  SPAIR3   :    T
** CHECKI ** YDEN(03)      /  SPAIR3   :    T
** CHECKI ** YDEN(04)      /  SPAIR3   :    T
** CHECKI ** ACTCO         /  SPDEP1   :    T
** CHECKI ** AWAT          /  SPDEP1   :    T
** CHECKI ** CONCRA        /  SPDEP1   :    T
** CHECKI ** DELT          /  SPDEP1   :    T
** CHECKI ** DMIX          /  SPDEP1   :    T
** CHECKI ** DUNSAT        /  SPDEP1   :    T
** CHECKI ** FACTCO        /  SPDEP1   :    T
** CHECKI ** FRACGP        /  SPDEP1   :    T
** CHECKI ** KPRIME        /  SPDEP1   :    T
** CHECKI ** KSUBSO        /  SPDEP1   :    T
** CHECKI ** MPCH2O        /  SPDEP1   :    T
** CHECKI ** NPRCP         /  SPDEP1   :    T
** CHECKI ** O2CONC        /  SPDEP1   :    T
** CHECKI ** O2DIFF        /  SPDEP1   :    T
** CHECKI ** PYRITE        /  SPDEP1   :    T
** CHECKI ** RAINGP        /  SPDEP1   :    T
** CHECKI ** RALAM         /  SPDEP1   :    T
** CHECKI ** RHOTAL        /  SPDEP1   :    T
** CHECKI ** RNLAM         /  SPDEP1   :    T
** CHECKI ** SULPRD        /  SPDEP1   :    T
** CHECKI ** THLAM         /  SPDEP2   :    T
** CHECKI ** U02TON        /  SPDEP2   :    T
** CHECKI ** UPERTH        /  SPDEP2   :    T
** CHECKI ** VLAKE         /  SPDEP2   :    T
** CHECKI ** BETA          /  SPLAKE   :    T
** CHECKI ** BF            /  SPLAKE   :    T
** CHECKI ** DCFING        /  SPLAKE   :    T
** CHECKI ** FFISH         /  SPLAKE   :    T
** CHECKI ** FISHIN        /  SPLAKE   :    T
** CHECKI ** FWAT          /  SPLAKE   :    T
** CHECKI ** KSUBDR        /  SPLAKE   :    T
** CHECKI ** TVOID         /  SPLAKE   :    T
** CHECKI ** WATIN         /  SPLAKE   :    T
** CHECKI ** XK1           /  SPLAKE   :    T
** CHECKI ** XK3           /  SPLAKE   :    T
** CHECKI ** AEXPT         /  SPRATH   :    T
** CHECKI ** ATAIL         /  SPRATH   :    T
** CHECKI ** DISTCO        /  SPRATH   :    T
** CHECKI ** EMANAT        /  SPRATH   :    T
** CHECKI ** FPREC         /  SPRATH   :    T
** CHECKI ** FWIND1        /  SPRATH   :    T
** CHECKI ** MOLTOB        /  SPRATH   :    T
** CHECKI ** RAINTH        /  SPRATH   :    T
** CHECKI ** SILT          /  SPRATH   :    T
** CHECKI ** THDISS        /  SPRATH   :    T
** CHECKI ** THFRAC        /  SPRATH   :    T
** CHECKI ** TSTEP         /  SPTIME   :    T
** CHECKI ** ACT14         /  CPALL    :    T
** CHECKI ** ACT34         /  CPALL    :    T
** CHECKI ** ACT94         /  CPALL    :    T
** CHECKI ** AEQBM         /  CPALL    :    T
** CHECKI ** CRAGIN        /  CPALL    :    T
** CHECKI ** CSINIT        /  CPALL    :    T
** CHECKI ** CTH230        /  CPALL    :    T
** CHECKI ** CTH232        /  CPALL    :    T
** CHECKI ** DIFLEN        /  CPALL    :    T
** CHECKI ** BYPBLK        /  CPALL    :    T
** CHECKI ** KRATE         /  CPALL    :    T
** CHECKI ** RADIUM        /  CPALL    :    T
** CHECKI ** RASIMM        /  CPALL    :    T
** CHECKI ** SFLUX         /  CPALL    :    T
** CHECKI ** TFINAL        /  CPALL    :    T
** CHECKI ** THINIT        /  CPALL    :    T
** CHECKI ** THINTH        /  CPALL    :    T
** CHECKI ** XLAML         /  CPALL    :    T
** CHECKI ** DSFISH(01)    /  CQFISH   :    T
** CHECKI ** DSFISH(02)    /  CQFISH   :    T
** CHECKI ** DSFISH(03)    /  CQFISH   :    T
** CHECKI ** DSFISH(04)    /  CQFISH   :    T
** CHECKI ** DSFISH(05)    /  CQFISH   :    T
** CHECKI ** DSFISH(06)    /  CQFISH   :    T
** CHECKI ** DSFISH(07)    /  CQFISH   :    T
** CHECKI ** DSFISH(08)    /  CQFISH   :    T
** CHECKI ** DSFISH(09)    /  CQFISH   :    T
** CHECKI ** DSFISH(10)    /  CQFISH   :    T
** CHECKI ** DSFISH(11)    /  CQFISH   :    T
** CHECKI ** MAXFDS        / CQFISH  :  T
** CHECKI ** TMXFDS        / CQFISH  :  T
** CHECKI ** AVGFDS        / CQFISH  :  T
** CHECKI ** DSMEAT(01)    / CQMEAT  :  T
** CHECKI ** DSMEAT(02)    / CQMEAT  :  T
** CHECKI ** DSMEAT(03)    / CQMEAT  :  T
** CHECKI ** DSMEAT(04)    / CQMEAT  :  T
** CHECKI ** DSMEAT(05)    / CQMEAT  :  T
** CHECKI ** DSMEAT(06)    / CQMEAT  :  T
** CHECKI ** DSMEAT(07)    / CQMEAT  :  T
** CHECKI ** DSMEAT(08)    / CQMEAT  :  T
** CHECKI ** DSMEAT(09)    / CQMEAT  :  T
** CHECKI ** DSMEAT(10)    / CQMEAT  :  T
** CHECKI ** DSMEAT(11)    / CQMEAT  :  T
** CHECKI ** MAXMDS        / CQMEAT  :  T
** CHECKI ** TMXMDS        / CQMEAT  :  T
** CHECKI ** AVGMDS        / CQMEAT  :  T
** CHECKI ** DSIRN(01)     / CQRND   :  T
** CHECKI ** DSIRN(02)     / CQRND   :  T
** CHECKI ** DSIRN(03)     / CQRND   :  T
** CHECKI ** DSIRN(04)     / CQRND   :  T
** CHECKI ** DSIRN(05)     / CQRND   :  T
** CHECKI ** DSIRN(06)     / CQRND   :  T
** CHECKI ** DSIRN(07)     / CQRND   :  T
** CHECKI ** DSIRN(08)     / CQRND   :  T
** CHECKI ** DSIRN(09)     / CQRND   :  T
** CHECKI ** DSIRN(10)     / CQRND   :  T
** CHECKI ** DSIRN(11)     / CQRND   :  T
** CHECKI ** MAXRND        / CQRND   :  T
** CHECKI ** TMXRND        / CQRND   :  T
** CHECKI ** AVGRND        / CQRND   :  T
** CHECKI ** DSTVEG(01)    / CQVEG   :  T
** CHECKI ** DSTVEG(02)    / CQVEG   :  T
** CHECKI ** DSTVEG(03)    / CQVEG   :  T
** CHECKI ** DSTVEG(04)    / CQVEG   :  T
** CHECKI ** DSTVEG(05)    / CQVEG   :  T
** CHECKI ** DSTVEG(06)    / CQVEG   :  T
** CHECKI ** DSTVEG(07)    / CQVEG   :  T
** CHECKI ** DSTVEG(08)    / CQVEG   :  T
** CHECKI ** DSTVEG(09)    / CQVEG   :  T
** CHECKI ** DSTVEG(10)    / CQVEG   :  T
** CHECKI ** DSTVEG(11)    / CQVEG   :  T
** CHECKI ** MAXVDS        / CQVEG   :  T
** CHECKI ** TMXVDS        / CQVEG   :  T
** CHECKI ** AVGVDS        / CQVEG   :  T
** CHECKI ** DSWATR(01)    / CQWAT   :  T
** CHECKI ** DSWATR(02)    / CQWAT   :  T
** CHECKI ** DSWATR(03)    / CQWAT   :  T
** CHECKI ** DSWATR(04)    / CQWAT   :  T
** CHECKI ** DSWATR(05)    / CQWAT   :  T
** CHECKI ** DSWATR(06)    / CQWAT   :  T
** CHECKI ** DSWATR(07)    / CQWAT   :  T
** CHECKI ** DSWATR(08)    / CQWAT   :  T
** CHECKI ** DSWATR(09)    / CQWAT   :  T
** CHECKI ** DSWATR(10)    / CQWAT   :  T
```

```
 ** CHECKI ** DSWATR(11)    /  CQWAT    :    T
 ** CHECKI ** MAXWDS        /  CQWAT    :    T
 ** CHECKI ** TMXWDS        /  CQWAT    :    T
 ** CHECKI ** AVGWDS        /  CQWAT    :    T
 ** CHECKI ** DSDUST(01)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(02)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(03)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(04)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(05)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(06)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(07)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(08)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(09)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(10)    /  CQXIN    :    T
 ** CHECKI ** DSDUST(11)    /  CQXIN    :    T
 ** CHECKI ** MAXXIN        /  CQXIN    :    T
 ** CHECKI ** TMXXIN        /  CQXIN    :    T
 ** CHECKI ** AVGXIN        /  CQXIN    :    T
 ** CHECKI ** DSTOT(01)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(02)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(03)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(04)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(05)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(06)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(07)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(08)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(09)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(10)     /  CQTDOS   :    T
 ** CHECKI ** DSTOT(11)     /  CQTDOS   :    T
 ** CHECKI ** MAXTOT        /  CQTDOS   :    T
 ** CHECKI ** TMXTOT        /  CQTDOS   :    T
 ** CHECKI ** AVGTOT        /  CQTDOS   :    T
 ** CHECKI ** LAKPH(01)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(02)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(03)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(04)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(05)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(06)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(07)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(08)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(09)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(10)     /  CQLAPH   :    T
 ** CHECKI ** LAKPH(11)     /  CQLAPH   :    T
 ** CHECKI ** MAXLPH        /  CQLAPH   :    T
 ** CHECKI ** TMXLPH        /  CQLAPH   :    T
 ** CHECKI ** AVGLPH        /  CQLAPH   :    T
1CHAINS:  (ONE PER LINE.  ''='' INDICATES SECULAR EQUILIBRIUM WITH PREVIOUS NUCLIDE.)



    77 SELECTED PARAMETERS VARY
    30 SELECTED PARAMETERS ARE CONSTANT
   112 CONSEQUENCES

 RECORD LENGTH =   190 WORDS.
```

```
1NUCLIDE ACCEPTANCE:

 NUCLIDE      # RUNS   % OF ACCEPTED RUNS

1 CASE FINISHED AT RUN NUMBER:     500, WITH:
    500 ACCEPTED RUNS,
      0 REJECTED RUNS, AND
      0 RUNS ON HOLD.

 CASE ENDED AT  14-MAY-86  19:41:35
 CPU TIME   0 HR   0.00 MIN




 STATISTICAL SUMMARY  -  500RUNS
 *******************


  NO  PARAMETER         MEAN          STD          GM            GSD
                                                 (SKEW)       (KURTOSIS
   5  BETA           0.1867E+00   0.4910E-01   0.1805E+00   0.1299E+01
                                              ( 0.5276E+00) ( 0.2619E+01)
   6  BF             0.3405E-04   0.5294E-04   0.1733E-04   0.3220E+01
                                              ( 0.5087E+01) ( 0.3985E+02)
   7  CONCRA         0.1053E+02   0.2120E+01   0.1029E+02   0.1243E+01
                                              (-0.3471E+00) ( 0.2273E+01)
  11  DEPVEL         0.1636E-01   0.7617E-02   0.1415E-01   0.1806E+01
                                              (-0.3816E-01) ( 0.1686E+01)
  12  DISTCO         0.5447E-01   0.2650E-01   0.4660E-01   0.1836E+01
                                              (-0.1252E-01) ( 0.1730E+01)
  15  DUREXP(01)     0.9965E+08   0.5913E+07   0.9947E+08   0.1061E+01
                                              ( 0.1015E+00) ( 0.1738E+01)
  16  DUREXP(02)     0.2892E+07   0.1133E+06   0.2890E+07   0.1040E+01
                                              ( 0.1482E+00) ( 0.1819E+01)
  17  DUREXP(03)     0.5157E+07   0.2615E+06   0.5150E+07   0.1052E+01
                                              (-0.5094E-01) ( 0.1749E+01)
  18  DUREXP(04)     0.5177E+07   0.2584E+06   0.5171E+07   0.1051E+01
                                              (-0.1301E+00) ( 0.1781E+01)
  19  EMANAT         0.3742E+00   0.1953E+00   0.3175E+00   0.1860E+01
                                              ( 0.4412E+00) ( 0.2328E+01)
  20  F              0.4305E+00   0.1039E+00   0.4177E+00   0.1284E+01
                                              ( 0.1952E+00) ( 0.2401E+01)
  21  FACTCD         0.3325E+00   0.6170E-01   0.3267E+00   0.1206E+01
                                              ( 0.2435E+00) ( 0.2386E+01)
  22  FAUT(01)       0.2489E+00   0.9215E-01   0.2314E+00   0.1481E+01
                                              ( 0.3938E+00) ( 0.2400E+01)
  23  FAUT(02)       0.2491E+00   0.9199E-01   0.2310E+00   0.1499E+01
                                              ( 0.3008E+00) ( 0.2406E+01)
  24  FBI            0.1358E-08   0.5993E-09   0.1203E-08   0.1695E+01
                                              ( 0.6969E-01) ( 0.1955E+01)
  25  FDEPED(01)     0.9000E+00   0.5888E-01   0.8981E+00   0.1063E+01
                                              (-0.8862E-02) ( 0.1780E+01)
  26  FDEPED(02)     0.8993E+00   0.5640E-01   0.8976E+00   0.1065E+01
                                              ( 0.2094E-01) ( 0.1833E+01)
```

```
27  FDEPED(03)     0.9002E+00   0.6024E-01   0.8982E+00   0.1069E+01
                                            (-0.1074E-01) ( 0.1723E+01)
28  FDEPED(04)     0.9982E-01   0.2897E-01   0.9531E-01   0.1367E+01
                                            (-0.5051E-01) ( 0.1833E+01)
29  FDEPSF(01)     0.8992E+00   0.5762E-01   0.8973E+00   0.1066E+01
                                            (-0.1102E-01) ( 0.1814E+01)
30  FDEPSF(02)     0.4593E+00   0.2334E+00   0.3764E+00   0.2105E+01
                                            ( 0.7387E-01) ( 0.2216E+01)
31  FDEPSF(03)     0.1997E+00   0.5762E-01   0.1910E+00   0.1357E+01
                                            ( 0.5776E-01) ( 0.1828E+01)
32  FDEPSF(04)     0.2043E+00   0.5528E-01   0.1963E+00   0.1339E+01
                                            (-0.1081E+00) ( 0.1885E+01)
35  FISHIN         0.8062E+04   0.2259E+04   0.7729E+04   0.1346E+01
                                            ( 0.3677E-02) ( 0.1800E+01)
36  FLOC(03)       0.3350E+00   0.1031E+00   0.3182E+00   0.1393E+01
                                            ( 0.2026E+00) ( 0.2498E+01)
37  FLOC(04)       0.3388E+00   0.1033E+00   0.3217E+00   0.1399E+01
                                            ( 0.8918E-01) ( 0.2446E+01)
38  FHEAT          0.4884E+00   0.2993E+00   0.3400E+00   0.3028E+01
                                            ( 0.2532E-01) ( 0.1726E+01)
39  FPREC          0.1230E+03   0.8107E+00   0.1230E+03   0.1007E+01
                                            ( 0.3748E-01) ( 0.2480E+01)
41  FRACV(03)      0.7523E+00   0.1507E+00   0.7368E+00   0.1229E+01
                                            (-0.1354E-01) ( 0.1719E+01)
42  FRACV(04)      0.7432E+00   0.1463E+00   0.7288E+00   0.1221E+01
                                            ( 0.1540E+00) ( 0.1840E+01)
43  FWAT           0.6291E+00   0.1036E+00   0.6205E+00   0.1182E+01
                                            ( 0.1324E+00) ( 0.2287E+01)
44  FWIND1         0.3501E+02   0.3010E+00   0.3501E+02   0.1009E+01
                                            (-0.7985E-01) ( 0.2818E+01)
45  FWIND2         0.1000E+00   0.8052E-03   0.1000E+00   0.1008E+01
                                            ( 0.5225E-01) ( 0.2836E+01)
46  HSRC           0.1986E+01   0.5720E+00   0.1899E+01   0.1358E+01
                                            ( 0.1992E-01) ( 0.1816E+01)
47  KPRIME         0.5863E-01   0.2935E-01   0.5261E-01   0.1583E+01
                                            ( 0.1559E+01) ( 0.6251E+01)
48  KSUBDR         0.1931E-03   0.3006E-03   0.9211E-04   0.3483E+01
                                            ( 0.4395E+01) ( 0.2990E+02)
51  MPCH2O         0.1487E+02   0.4177E+01   0.1422E+02   0.1363E+01
                                            (-0.7054E-01) ( 0.2348E+01)
52  NPRCP          0.3999E+00   0.3673E-02   0.3999E+00   0.1009E+01
                                            (-0.1207E-01) ( 0.2818E+01)
54  O2DIFF         0.7652E+02   0.1982E+02   0.7389E+02   0.1310E+01
                                            ( 0.2457E+00) ( 0.2566E+01)
55  OF             0.6336E+00   0.1013E+00   0.6255E+00   0.1175E+01
                                            ( 0.2033E+00) ( 0.2530E+01)
57  RAINSP         0.2756E+00   0.1191E+00   0.2485E+00   0.1611E+01
                                            ( 0.4314E+00) ( 0.2497E+01)
58  RAINTH         0.9011E+00   0.2017E-01   0.9008E+00   0.1023E+01
                                            (-0.4039E-02) ( 0.2459E+01)
60  RHOTAL         0.1395E+07   0.1003E+06   0.1391E+07   0.1075E+01
                                            (-0.8735E-01) ( 0.2967E+01)
63  SULPRD         0.1498E-05   0.5890E-06   0.1390E-05   0.1475E+01
                                            ( 0.9829E+00) ( 0.4311E+01)
64  TMDISS         0.2028E-01   0.5765E-02   0.1939E-01   0.1362E+01
```

```
                                                 (-0.9914E-01) ( 0.1881E+01)
 65  THFRAC          0.3011E+00   0.4007E-01   0.2984E+00   0.1146E+01
                                                 (-0.4797E-01) ( 0.2424E+01)
 68  TVOID           0.3987E+00   0.2107E-01   0.3981E+00   0.1054E+01
                                                 ( 0.2769E-01) ( 0.2247E+01)
 71  UWIND           0.5300E+01   0.2180E-01   0.5300E+01   0.1004E+01
                                                 (-0.3041E+00) ( 0.3049E+01)
 72  VDISP           0.6016E+02   0.4592E+01   0.5998E+02   0.1079E+01
                                                 ( 0.2179E+00) ( 0.2819E+01)
 75  VEGLAM(01)      0.3032E-08   0.5658E-09   0.2978E-08   0.1212E+01
                                                 (-0.3932E-01) ( 0.1846E+01)
 76  VEGLAM(02)      0.4994E-06   0.5616E-07   0.4962E-06   0.1120E+01
                                                 (-0.5776E-03) ( 0.1820E+01)
 77  VEGLAM(03)      0.5019E-06   0.5819E-07   0.4985E-06   0.1124E+01
                                                 (-0.9790E-02) ( 0.1815E+01)
 78  VEGLAM(04)      0.5011E-06   0.5637E-07   0.4979E-06   0.1121E+01
                                                 (-0.5601E-01) ( 0.1815E+01)
 83  XK1             0.4875E+00   0.2681E+00   0.4236E+00   0.1699E+01
                                                 ( 0.1117E+01) ( 0.3789E+01)
 84  XK3             0.2220E-01   0.3587E-02   0.2191E-01   0.1174E+01
                                                 ( 0.3646E+00) ( 0.2421E+01)
 86  YDEN(01)        0.5028E+03   0.5780E+02   0.4994E+03   0.1124E+01
                                                 (-0.9842E-01) ( 0.1824E+01)
 87  YDEN(02)        0.7534E+03   0.1402E+03   0.7399E+03   0.1212E+01
                                                 (-0.5490E-01) ( 0.1881E+01)
 88  YDEN(03)        0.1984E+04   0.2832E+03   0.1964E+04   0.1155E+01
                                                 ( 0.5129E-01) ( 0.1849E+01)
 89  YDEN(04)        0.2012E+04   0.2862E+03   0.1991E+04   0.1157E+01
                                                 (-0.1241E+00) ( 0.1847E+01)
 90  ACT14           0.9457E+00   0.1359E-06   0.9457E+00   0.1000E+01
                                                 (-0.1658E+08) ( 0.2701E+15)
 91  ACT34           0.8459E+00   0.1440E-06   0.8459E+00   ???????????
                                                 (-0.1040E+08) ( 0.9673E+14)
 92  ACT94           0.6053E+00   0.0000E+00   0.6053E+00   ???????????
                                                 (+++++++++++) (-----------)
 93  AE2BM           0.1558E+02   0.2960E+01   0.1530E+02   0.1206E+01
                                                 ( 0.5102E+00) ( 0.2700E+01)
 94  CRAGIN          0.4852E-06   0.2425E-06   0.4257E-06   0.1702E+01
                                                 ( 0.8295E+00) ( 0.3425E+01)
 95  CSINIT          0.8715E+02   0.2038E+02   0.8481E+02   0.1265E+01
                                                 ( 0.5372E+00) ( 0.3342E+01)
 96  CIM230          0.8773E-04   0.1835E-04   0.8566E-04   0.1251E+01
                                                 (-0.2129E+00) ( 0.2376E+01)
 97  CIM232          0.2114E+01   0.1520E+00   0.2106E+01   0.1075E+01
                                                 (-0.8735E-01) ( 0.2967E+01)
 98  DIFLEN          0.3579E+01   0.1951E+01   0.3098E+01   0.1725E+01
                                                 ( 0.1059E+01) ( 0.3844E+01)
 99  BYPBLK          0.5129E+03   0.3689E+02   0.5116E+03   0.1075E+01
                                                 (-0.8735E-01) ( 0.2967E+01)
100  KRATE           0.2387E-01   0.5588E-02   0.2323E-01   0.1265E+01
                                                 ( 0.5330E+00) ( 0.3320E+01)
101  RADIUM          0.1755E-05   0.3669E-06   0.1713E-05   0.1251E+01
                                                 (-0.2129E+00) ( 0.2376E+01)
102  RASINM          0.1269E-05   0.3374E-06   0.1223E-05   0.1322E+01
                                                 ( 0.2225E+00) ( 0.2587E+01)
```

```
103  SFLUX        0.3485E+02   0.8158E+01   0.3391E+02   0.1265E+01
                                          ( 0.5330E+00) ( 0.3320E+01)
104  TFINAL       0.1453E+03   0.2540E+02   0.1431E+03   0.1192E+01
                                          ( 0.2943E+00) ( 0.2640E+01)
105  THINIT       0.2114E+01   0.1520E+00   0.2108E+01   0.1075E+01
                                          (-0.8736E-01) ( 0.2967E+01)
106  THINTH       0.4155E-04   0.8366E-05   0.4063E-04   0.1243E+01
                                          (-0.3471E+00) ( 0.2273E+01)
107  XLAML        0.2001E+01   0.1838E-01   0.2001E+01   0.1009E+01
                                          ( 0.6218E-01) ( 0.2827E+01)
108  DSFISH(01)   0.6103E-05   0.1150E-04   0.2730E-05   0.3561E+01
                                          ( 0.6376E+01) ( 0.5972E+02)
109  DSFISH(02)   0.5718E-05   0.1188E-04   0.2465E-05   0.3612E+01
                                          ( 0.6852E+01) ( 0.6165E+02)
110  DSFISH(03)   0.9023E-05   0.2223E-04   0.3375E-05   0.3924E+01
                                          ( 0.8444E+01) ( 0.9615E+02)
111  DSFISH(04)   0.2928E-04   0.9102E-04   0.7610E-05   0.4900E+01
                                          ( 0.1030E+02) ( 0.1431E+03)
112  DSFISH(05)   0.3543E-04   0.7999E-04   0.1124E-04   0.4561E+01
                                          ( 0.6765E+01) ( 0.6838E+02)
113  DSFISH(06)   0.2783E-04   0.6415E-04   0.9077E-05   0.4401E+01
                                          ( 0.6533E+01) ( 0.6055E+02)
114  DSFISH(07)   0.2021E-04   0.4093E-04   0.6937E-05   0.4378E+01
                                          ( 0.4698E+01) ( 0.2998E+02)
115  DSFISH(08)   0.1588E-04   0.3196E-04   0.5443E-05   0.4468E+01
                                          ( 0.4783E+01) ( 0.3096E+02)
116  DSFISH(09)   0.1308E-04   0.2667E-04   0.4408E-05   0.4571E+01
                                          ( 0.5006E+01) ( 0.3409E+02)
117  DSFISH(10)   0.1108E-04   0.2299E-04   0.3667E-05   0.4665E+01
                                          ( 0.5296E+01) ( 0.3843E+02)
118  DSFISH(11)   0.9586E-05   0.2024E-04   0.3120E-05   0.4745E+01
                                          ( 0.5602E+01) ( 0.4334E+02)
119  MAXFDS       0.4796E-04   0.1147E-03   0.1389E-04   0.4823E+01
                                          ( 0.6792E+01) ( 0.6625E+02)
120  TMXFDS       0.1690E+03   0.4577E+02   0.1401E+03   0.2855E+01
                                          (-0.1856E+01) ( 0.8395E+01)
121  AVGFDS       0.1745E-04   0.3535E-04   0.6592E-05   0.4001E+01
                                          ( 0.5778E+01) ( 0.4916E+02)
122  DSMEAT(01)   0.1763E-08   0.2247E-08   0.8146E-09   0.4283E+01
                                          ( 0.3130E+01) ( 0.1896E+02)
123  DSMEAT(02)   0.1740E-08   0.2216E-08   0.8039E-09   0.4282E+01
                                          ( 0.3132E+01) ( 0.1900E+02)
124  DSMEAT(03)   0.1717E-08   0.2187E-08   0.7934E-09   0.4282E+01
                                          ( 0.3135E+01) ( 0.1904E+02)
125  DSMEAT(04)   0.1694E-08   0.2158E-08   0.7831E-09   0.4281E+01
                                          ( 0.3137E+01) ( 0.1909E+02)
126  DSMEAT(05)   0.1672E-08   0.2129E-08   0.7731E-09   0.4280E+01
                                          ( 0.3140E+01) ( 0.1913E+02)
127  DSMEAT(06)   0.1651E-08   0.2101E-08   0.7632E-09   0.4280E+01
                                          ( 0.3142E+01) ( 0.1918E+02)
128  DSMEAT(07)   0.1629E-09   0.2074E-08   0.7536E-09   0.4279E+01
                                          ( 0.3145E+01) ( 0.1922E+02)
129  DSMEAT(08)   0.1609E-08   0.2047E-08   0.7441E-09   0.4278E+01
                                          ( 0.3147E+01) ( 0.1927E+02)
130  DSMEAT(09)   0.1588E-08   0.2021E-08   0.7349E-09   0.4278E+01
```

```
                                                  ( 0.3150E+01) ( 0.1931E+02)
131  DSMEAT(10)      0.1569E-08   0.1995E-08   0.7258E-09   0.4277E+01
                                                  ( 0.3152E+01) ( 0.1936E+02)
132  DSMEAT(11)      0.1549E-08   0.1970E-08   0.7170E-09   0.4276E+01
                                                  ( 0.3155E+01) ( 0.1941E+02)
133  MAXMDS          0.1763E-08   0.2247E-08   0.8146E-09   0.4283E+01
                                                  ( 0.3130E+01) ( 0.1896E+02)
134  TMXMDS          0.0000E+00   0.0000E+00   0.1000E+01   0.1000E+01
                                                  (???????????) (???????????)
135  AVGMDS          0.1652E-08   0.2104E-08   0.7641E-09   0.4280E+01
                                                  ( 0.3142E+01) ( 0.1917E+02)
136  DSIRN(01)       0.1082E-04   0.9920E-05   0.7546E-05   0.2435E+01
                                                  ( 0.2248E+01) ( 0.9398E+01)
137  DSIRN(02)       0.9534E-05   0.9359E-05   0.6390E-05   0.2547E+01
                                                  ( 0.2409E+01) ( 0.1035E+02)
138  DSIRN(03)       0.8917E-05   0.8871E-05   0.5955E-05   0.2550E+01
                                                  ( 0.2484E+01) ( 0.1089E+02)
139  DSIRN(04)       0.7936E-05   0.7801E-05   0.5389E-05   0.2491E+01
                                                  ( 0.2534E+01) ( 0.1133E+02)
140  DSIRN(05)       0.7432E-05   0.7137E-05   0.5124E-05   0.2451E+01
                                                  ( 0.2618E+01) ( 0.1267E+02)
141  DSIRN(06)       0.7305E-05   0.6969E-05   0.5047E-05   0.2447E+01
                                                  ( 0.2597E+01) ( 0.1266E+02)
142  DSIRN(07)       0.7210E-05   0.6874E-05   0.4983E-05   0.2446E+01
                                                  ( 0.2592E+01) ( 0.1262E+02)
143  DSIRN(08)       0.7118E-05   0.6780E-05   0.4920E-05   0.2446E+01
                                                  ( 0.2588E+01) ( 0.1257E+02)
144  DSIRN(09)       0.7028E-05   0.6689E-05   0.4859E-05   0.2445E+01
                                                  ( 0.2583E+01) ( 0.1253E+02)
145  DSIRN(10)       0.6939E-05   0.6600E-05   0.4799E-05   0.2445E+01
                                                  ( 0.2579E+01) ( 0.1249E+02)
146  DSIRN(11)       0.6853E-05   0.6512E-05   0.4741E-05   0.2444E+01
                                                  ( 0.2575E+01) ( 0.1245E+02)
147  MAXRND          0.1082E-04   0.9920E-05   0.7546E-05   0.2435E+01
                                                  ( 0.2248E+01) ( 0.9398E+01)
148  TMXRND          0.0000E+00   0.0000E+00   0.1000E+01   0.1000E+01
                                                  (???????????) (???????????)
149  AVGRND          0.7804E-05   0.7477E-05   0.5353E-05   0.2472E+01
                                                  ( 0.2505E+01) ( 0.1160E+02)
150  DSTVEG(01)      0.2719E-05   0.2424E-05   0.1974E-05   0.2272E+01
                                                  ( 0.2775E+01) ( 0.1614E+02)
151  DSTVEG(02)      0.2682E-05   0.2389E-05   0.1948E-05   0.2271E+01
                                                  ( 0.2769E+01) ( 0.1607E+02)
152  DSTVEG(03)      0.2647E-05   0.2355E-05   0.1922E-05   0.2270E+01
                                                  ( 0.2763E+01) ( 0.1600E+02)
153  DSTVEG(04)      0.2611E-05   0.2321E-05   0.1897E-05   0.2269E+01
                                                  ( 0.2757E+01) ( 0.1593E+02)
154  DSTVEG(05)      0.2577E-05   0.2288E-05   0.1873E-05   0.2268E+01
                                                  ( 0.2750E+01) ( 0.1587E+02)
155  DSTVEG(06)      0.2543E-05   0.2256E-05   0.1849E-05   0.2267E+01
                                                  ( 0.2744E+01) ( 0.1580E+02)
156  DSTVEG(07)      0.2510E-05   0.2225E-05   0.1826E-05   0.2267E+01
                                                  ( 0.2738E+01) ( 0.1573E+02)
157  DSTVEG(08)      0.2478E-05   0.2194E-05   0.1803E-05   0.2266E+01
                                                  ( 0.2731E+01) ( 0.1566E+02)
```

```
158  DSTVEG(09)      0.2447E-05  0.2164E-05  0.1781E-05  0.2265E+01
                                        ( 0.2725E+01) ( 0.1560E+02)
159  DSTVEG(10)      0.2416E-05  0.2135E-05  0.1759E-05  0.2264E+01
                                        ( 0.2719E+01) ( 0.1553E+02)
160  DSTVEG(11)      0.2386E-05  0.2106E-05  0.1737E-05  0.2263E+01
                                        ( 0.2712E+01) ( 0.1546E+02)
161  MAXVDS          0.2719E-05  0.2424E-05  0.1974E-05  0.2272E+01
                                        ( 0.2775E+01) ( 0.1614E+02)
162  TMXVDS          0.0000E+00  0.0000E+00  0.1000E+01  0.1000E+01
                                        (???????????) (???????????)
163  AVGVDS          0.2546E-05  0.2259E-05  0.1851E-05  0.2268E+01
                                        ( 0.2745E+01) ( 0.1581E+02)
164  DSWATR(01)      0.8385E-05  0.4021E-05  0.7589E-05  0.1553E+01
                                        ( 0.1443E+01) ( 0.5424E+01)
165  DSWATR(02)      0.7721E-05  0.4395E-05  0.6851E-05  0.1602E+01
                                        ( 0.2782E+01) ( 0.1826E+02)
166  DSWATR(03)      0.1232E-04  0.1345E-04  0.9381E-05  0.1958E+01
                                        ( 0.6167E+01) ( 0.6281E+02)
167  DSWATR(04)      0.3817E-04  0.5637E-04  0.2115E-04  0.2786E+01
                                        ( 0.4551E+01) ( 0.3451E+02)
168  DSWATR(05)      0.5336E-04  0.7747E-04  0.3123E-04  0.2669E+01
                                        ( 0.5005E+01) ( 0.4088E+02)
169  DSWATR(06)      0.3864E-04  0.4785E-04  0.2523E-04  0.2444E+01
                                        ( 0.4626E+01) ( 0.3398E+02)
170  DSWATR(07)      0.2795E-04  0.2890E-04  0.1928E-04  0.2395E+01
                                        ( 0.3863E+01) ( 0.2942E+02)
171  DSWATR(08)      0.2184E-04  0.2125E-04  0.1513E-04  0.2464E+01
                                        ( 0.3383E+01) ( 0.2303E+02)
172  DSWATR(09)      0.1791E-04  0.1704E-04  0.1225E-04  0.2547E+01
                                        ( 0.2816E+01) ( 0.1541E+02)
173  DSWATR(10)      0.1515E-04  0.1433E-04  0.1019E-04  0.2616E+01
                                        ( 0.2469E+01) ( 0.1161E+02)
174  DSWATR(11)      0.1311E-04  0.1246E-04  0.8673E-05  0.2668E+01
                                        ( 0.2340E+01) ( 0.1068E+02)
175  MAXWDS          0.7228E-04  0.1058E-03  0.3861E-04  0.2914E+01
                                        ( 0.3514E+01) ( 0.1859E+02)
176  TMXWDS          0.1690E+03  0.4577E+02  0.1401E+03  0.2855E+01
                                        (-0.1856E+01) ( 0.8395E+01)
177  AVGWDS          0.2435E-04  0.2176E-04  0.1832E-04  0.2086E+01
                                        ( 0.2937E+01) ( 0.1660E+02)
178  DSDUST(01)      0.9917E-06  0.3109E-06  0.9421E-06  0.1387E+01
                                        ( 0.4284E+00) ( 0.2964E+01)
179  DSDUST(02)      0.9784E-06  0.3060E-06  0.9297E-06  0.1386E+01
                                        ( 0.4271E+00) ( 0.2963E+01)
180  DSDUST(03)      0.9654E-06  0.3012E-06  0.9176E-06  0.1385E+01
                                        ( 0.4258E+00) ( 0.2963E+01)
181  DSDUST(04)      0.9527E-06  0.2966E-06  0.9057E-06  0.1384E+01
                                        ( 0.4244E+00) ( 0.2962E+01)
182  DSDUST(05)      0.9402E-06  0.2920E-06  0.8941E-06  0.1383E+01
                                        ( 0.4231E+00) ( 0.2961E+01)
183  DSDUST(06)      0.9280E-06  0.2876E-06  0.8827E-06  0.1382E+01
                                        ( 0.4217E+00) ( 0.2960E+01)
184  DSDUST(07)      0.9161E-06  0.2832E-06  0.8715E-06  0.1381E+01
                                        ( 0.4203E+00) ( 0.2959E+01)
185  DSDUST(08)      0.9044E-06  0.2790E-06  0.8606E-06  0.1380E+01
```

```
                                              ( 0.4189E+00) ( 0.2958E+01)
186  DSDUST(09)      0.8930E-06  0.2748E-06  0.8499E-06   0.1379E+01
                                              ( 0.4174E+00) ( 0.2957E+01)
187  DSDUST(10)      0.8819E-06  0.2708E-06  0.8395E-06   0.1378E+01
                                              ( 0.4160E+00) ( 0.2956E+01)
188  DSDUST(11)      0.8708E-06  0.2668E-06  0.8292E-06   0.1377E+01
                                              ( 0.4145E+00) ( 0.2956E+01)
189  MAXXIN          0.9917E-06  0.3109E-06  0.9421E-06   0.1387E+01
                                              ( 0.4284E+00) ( 0.2964E+01)
190  TMXXIN          0.0000E+00  0.0000E+00  0.1000E+01   0.1000E+01
                                              (???????????) (???????????)
191  AVGXIN     .    0.9291E-06  0.2880E-06  0.8837E-06   0.1382E+01
                                              ( 0.4218E+00) ( 0.2960E+01)
192  DSTOT(01)       0.2902E-04  0.1704E-04  0.2552E-04   0.1636E+01
                                              ( 0.2719E+01) ( 0.1670E+02)
193  DSTOT(02)       0.2664E-04  0.1709E-04  0.2315E-04   0.1657E+01
                                              ( 0.3090E+01) ( 0.1806E+02)
194  DSTOT(03)       0.3387E-04  0.3095E-04  0.2754E-04   0.1805E+01
                                              ( 0.5155E+01) ( 0.4170E+02)
195  DSTOT(04)       0.7896E-04  0.1255E-03  0.4834E-04   0.2419E+01
                                              ( 0.6513E+01) ( 0.6579E+02)
196  DSTOT(05)       0.9975E-04  0.1292E-03  0.6346E-04   0.2421E+01
                                              ( 0.3920E+01) ( 0.2410E+02)
197  DSTOT(06)       0.7725E-04  0.9594E-04  0.5307E-04   0.2229E+01
                                              ( 0.5318E+01) ( 0.4887E+02)
198  DSTOT(07)       0.5881E-04  0.5839E-04  0.4326E-04   0.2104E+01
                                              ( 0.3040E+01) ( 0.1435E+02)
199  DSTOT(08)       0.4822E-04  0.4498E-04  0.3647E-04   0.2046E+01
                                              ( 0.3055E+01) ( 0.1497E+02)
200  DSTOT(09)       0.4135E-04  0.3746E-04  0.3167E-04   0.2020E+01
                                              ( 0.3128E+01) ( 0.1626E+02)
201  DSTOT(10)       0.3647E-04  0.3236E-04  0.2814E-04   0.2008E+01
                                              ( 0.3236E+01) ( 0.1801E+02)
202  DSTOT(11)       0.3280E-04  0.2865E-04  0.2546E-04   0.1997E+01
                                              ( 0.3350E+01) ( 0.1981E+02)
203  MAXTOT          0.1314E-03  0.1867E-03  0.7636E-04   0.2635E+01
                                              ( 0.3894E+01) ( 0.2351E+02)
204  TMXTOT          0.1603E+03  0.5637E+02  0.1122E+03   0.4179E+01
                                              (-0.1711E+01) ( 0.5885E+01)
205  AVGTOT          0.5308E-04  0.4775E-04  0.4137E-04   0.1946E+01
                                              ( 0.3442E+01) ( 0.2107E+02)
206  LAKPH(01)       0.3999E+01  0.2555E-01  0.3999E+01   0.1006E+01
                                              ( 0.4737E+00) ( 0.3795E+01)
207  LAKPH(02)       0.4159E+01  0.1147E-01  0.4159E+01   0.1003E+01
                                              ( 0.2209E+01) ( 0.1008E+02)
208  LAKPH(03)       0.4557E+01  0.4309E+00  0.4540E+01   0.1085E+01
                                              ( 0.3666E+01) ( 0.1665E+02)
209  LAKPH(04)       0.6409E+01  0.8674E+00  0.6343E+01   0.1160E+01
                                              (-0.1117E+01) ( 0.2654E+01)
210  LAKPH(05)       0.7119E+01  0.1976E+00  0.7116E+01   0.1032E+01
                                              (-0.6668E+01) ( 0.7226E+02)
211  LAKPH(06)       0.7223E+01  0.6015E-01  0.7222E+01   0.1009E+01
                                              (-0.5527E+01) ( 0.6020E+02)
212  LAKPH(07)       0.7252E+01  0.2562E-01  0.7252E+01   0.1004E+01
                                              (-0.5438E+01) ( 0.5573E+02)
```

```
213  LAKPH(08)      0.7262E+01   0.1262E-01   0.7262E+01   0.1002E+01
                                             (-0.6984E+01) ( 0.8265E+02)
214  LAKPH(09)      0.7266E+01   0.6655E-02   0.7266E+01   0.1001E+01
                                             (-0.9108E+01) ( 0.1257E+03)
215  LAKPH(10)      0.7268E+01   0.3689E-02   0.7268E+01   0.1001E+01
                                             (-0.1145E+02) ( 0.1786E+03)
216  LAKPH(11)      0.7269E+01   0.2126E-02   0.7269E+01   0.1000E+01
                                             (-0.1368E+02) ( 0.2274E+03)
217  MAXLPH         0.7269E+01   0.2126E-02   0.7269E+01   0.1000E+01
                                             (-0.1368E+02) ( 0.2274E+03)
218  TMXLPH         0.5000E+03   0.0000E+00   0.5000E+03   0.1000E+01
                                             (???????????) (???????????)
219  AVGLPH         0.6417E+01   0.1339E+00   0.6416E+01   0.1021E+01
                                             (-0.4952E+00) ( 0.3510E+01)
```

APPENDIX H. LISTINGS OF CODE DEVELOPED FOR THE URANIUM TAILINGS MODEL AS IMPLEMENTED IN SYVAC3

The interfaces of a system model to SYVAC3 were described in Section 2.4. Apart from the file of input data, (reproduced in Appendix G), there are three major interfaces to the SYVAC3 executive modules, involving common blocks, time-independent variables and time-dependent variables. These three interfaces, and the required subroutine and code, are as follows:

(1) COMMON BLOCKS - SYVAC3 uses user-defined common blocks to store the input parameter data that are required by the system models. For the uranium tailings system model, these input parameters are included in five "INCLUDE" files: SPAIR.INC, SPDEP.INC, SPLAKE.INP, SPRATH.INP and SPTIME.INP (the prefix "SP" is used to denote sampled parameters). INCLUDE files are also used to store automatically all major intermediate parameters and all major consequence parameters. These parameters are included in the files CPALL.INC ("CP" is used to denote the calculated parameters) and CQFISH.INC, CQMEAT.INC, CQRND.INC, CQVEG.INC, CQWAT.INC, CQXIN.INC, CQTDOS.INC and CQLAPH.INC ("CQ" is used to denote the consequence parameters). Three subroutines in SYVAC3 must also be modified to reference the above INCLUDE files. These subroutines are SELECT, INITCQ and WRDATA.

(2) TIME-INDEPENDENT CALCULATIONS - The uranium tailings system model contains a number of calculated variables that are constant within a given simulation, and that are required at different locations throughout the code. These variables are calculated in subroutine DEPPAR (and passed to other subroutines in CPALL.INC). DEPPAR also references subroutine BISECT which uses subroutine TFFUNC to calculate TFINAL.

(3) TIME-DEPENDENT CALCULATIONS - The bulk of the uranium tailings system model is contained within the subroutines that calculate time-dependent variables. The main interface to SYVAC3 is through subroutine SIMLAT. As shown in Figure 5, SIMLAT references six other subroutines: SIMMIX, SIMUNZ, SIMRAN, SIMWAT, SIMAIR and SIMDOS. Figures 6 to 10 show that the first five of these latter subroutines also reference a total of 19 subroutines, all prefixed with the letter "Z", that contain time-dependent components of the uranium tailings system model. Three of these 19 subroutines also use BISECT and a tailings-specific function. ZPHLAK uses PHFUNC (and BISECT) to calculate PHLAKE; ZDEPT1 uses DEPFNC to calculate DEPTH1, and ZFRACR uses FRCFNC to calculate FRACRA. Subroutine TFFUNC is also referenced by ZREPGY to calculate REPGYP. (TFFUNC is also referenced by BISECT from DEPPAR to calculate TFINAL). Finally, two subroutines were added to the SYVAC3 executive to store intermediate data (BGSAVE) and to calculate statistical output (BGSTAT) for probabilistic simulations.

The following page contains listings of all subroutines and code used to implement the uranium tailings model in SYVAC3. All developed code follows ANSI FORTRAN 77 standards, and has been compiled using the IBM Personal Computer FORTRAN Compiler[1] . The listings are in alphabetical order, and include:

1/ Ryan-McFarland Corporation, 1984. IBM Personal Computer Professional FORTRAN. Reference volume and Installation and Use volume, IBM Personal Computer Software, reference numbers 6138815 and 6362003.

Subroutines and INCLUDE files	Description	Page
BGSAVE	saves intermediate results	H-3
BGSTAT	performs statistical calculations	H-5
BISECT	finds roots of an equation	H-7
CPALL.INC	calculated time-independent variables	H-10
CQFISH.INC	dose from fish	H-12
CQLAPH.INC	lake pH	H-13
CQMEAT.INC	dose from caribou meat	H-14
CQRND.INC	dose from radon gas	H-15
CQTDOS.INC	total dose	H-16
CQVEG.INC	dose from vegetables	H-17
CQXIN.INC	dose from radium in dust	H-18
DEPFNC	used to calculate DEPTH1	H-19
DEPPAR	calculates time-independent parameters	H-20
D1COM.INC	parameters used by PHFUNC	H-22
FRCCOM.INC	parameters used by FRCFNC	H-23
FRCFNC	used to calculate FRACRA	H-24
INITCQ	initialize consequence parameters	H-26
PHCOM.INC	parameters used by PHFUNC	H-28
PHFUNC	used to calculate PHLAKE	H-29
SELECT	used to select sampled values	H-30
SIMAIR	air dispersion pathway	H-33
SIMDOS	calculation of total dose	H-36
SIMLAT	main SYVAC3 interface for tailings model	H-39
SIMMIX	mixing zone and lake concentrations	H-41
SIMRAN	radon and radon emission	H-43
SIMUNZ	unsaturated zone	H-45
SIMWAT	water pathways	H-47
SPAIR.INC	sampled parameters - air	H-49
SPDEP.INC	sampled parameters for DEPPAR and SIMMIX	H-51
SPLAKE.INC	sampled parameters - lake	H-53
SPRATH.INC	sampled parameters for SIMUNZ and SIMRAN	H-54
SPTIME.INC	sampled parameters - time values	H-55
TF.INC	parameters used by TFFUNC	H-56
TFFUNC	used to calculate TFINAL and REPGYP	H-57
WRDATA	output results	H-58
ZCALAK	calculates CALAKE	H-64
ZCAMIX	calculates CAMIX	H-67
ZCLAKE	calculates CLAKE	H-70
ZCONCC	calculates CONCCA	H-74
ZCONCS	calculates CONCS8	H-76
ZCRAAV	calculates CRAAVG	H-78
ZDEPT1	calculates DEPTH1	H-81
ZEMRAA	calculates EMRATE(RA)	H-83
ZEMRAN	calculates EMRATE(RN)	H-85
ZFRACR	calculates FRACRA	H-88
ZGYPLF	calculates GYPLFT	H-92
ZION	calculates ION	H-95
ZPHLAK	calculates PHLAKE	H-97
ZRACO1	calculates RACON1	H-100
ZRAFTH	calculates RAFTH	H-102
ZRAMIX	calculates RAMIX	H-104
ZREPGY	calculates REPGYP	H-107
ZSO4MI	calculates SO4MIX	H-109
ZSULAK	calculates SULAKE	H-112

```
      SUBROUTINE BGBAVE (VALUE)
C
C************************************************************** BGSAVE
C
C  Store sums etc. of sampled, calculated and consequence parameters
C        for later use in statistical calculations in BGSTAT
C
C  86-MAY-13  VERSION 01  B.GOODWIN
C
C**********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'H:PARDES.INC'
      INCLUDE 'H:PCOUNT.INC'
      INCLUDE 'H:PDISTN.INC'
C
      INTEGER I
C                 ! do loop index []
      INTEGER NPAR
C                 ! do loop index []
      INTEGER RUNNO
C                 ! current call to this routine. [-]
      DOUBLE PRECISION SUMX
C                 ! sum of parameter values [-]
      DOUBLE PRECISION SUMX2
C                 ! sum of square of parameter values [-]
      DOUBLE PRECISION SUMX3
C                 ! sum of cube of parameter values [-]
      DOUBLE PRECISION SUMX4
C                 ! sum of fourth power of parameter values [-]
      DOUBLE PRECISION SUMLX
C                 ! sum of ln(parameter values) [-]
      DOUBLE PRECISION SUMLX2
C                 ! sum of square of ln(parameter values) [-]
      DOUBLE PRECISION VALUE(MXPAR)
C                 ! parameter values generated in WRDATA [-]
      DOUBLE PRECISION X
C                 ! temporary storage
      COMMON /BGSAV/
     1 SUMX(250),
     1 SUMX2(250),
     1 SUMX3(250),
     1 SUMX4(250),
     1 SUMLX(250),
     1 SUMLX2(250),
     1 RUNNO
C                 ! common info used in BGSTAT
C
      LOGICAL FIRST
C                 ! Is this the first call to this subroutine?
C
      DATA FIRST /.TRUE./
C
      NPAR = NSPAR+NUMCQ
C
C  INITIALIZE ARRAYS
C
      IF (FIRST) THEN
```

```
         FIRST = .FALSE.
         DO 700 I=1,NPAR
            SUMX(I)  = 0.0D00
            SUMX2(I) = 0.0D00
            SUMX3(I) = 0.0D00
            SUMX4(I) = 0.0D00
            SUMLX(I) = 0.0D00
            SUMLX2(I)= 0.0D00
  700    CONTINUE
      ENDIF
C
C  SAVE PARAMETER VALUES
C
      RUNNO = RUNNO + 1
      DO 750 I=1,NPAR
         IF (PARTYP(I).NE.'LIT' .AND. DSTTPS(I).NE.'CONST'  ) THEN
            X = VALUE(I)
            SUMX(I)  = SUMX(I)  + X
            SUMX2(I) = SUMX2(I) + X*X
            SUMX3(I) = SUMX3(I) + X*X*X
            SUMX4(I) = SUMX4(I) + X**4
            IF (X .GT. 0.0D00) THEN
               X = LOG(X)
               SUMLX(I) = SUMLX(I) + X
               SUMLX2(I)= SUMLX2(I)+ X*X
            ENDIF
         ENDIF
  750 CONTINUE
      RETURN
      END
      SUBROUTINE BGSTAT (IULPT)
C
C************************************************************** BGSTAT
C
C  Calculates statistical stuff* for selected, calculated and
C      consequence parameters, using data saved by BGSTAT
C
C  86-MAY-13  VERSION 01  B.GOODWIN
C
C                *stuff=mean,std,gm,gsd,skewness,kurtosis
C*********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'H:PARDES.INC'
      INCLUDE 'H:PCOUNT.INC'
      INCLUDE 'H:PDISTN.INC'
C
C
      INTEGER I
C                ! do loop index []
      INTEGER IULPT
C                ! line printer unit number []
      INTEGER NPAR
C                ! do loop index []
      INTEGER RUNNUM
C                ! number of calls to BGSAVE and should equal the
C                ! number of accepted runs
C
      DOUBLE PRECISION GM
C                ! geometric mean
      DOUBLE PRECISION GSD
C                ! geometric standard deviation
      DOUBLE PRECISION KURT
C                ! kurtosis
      DOUBLE PRECISION MEAN
C                ! mean value
      DOUBLE PRECISION NOX
C                ! double precision equivalent of RUNNUM
      DOUBLE PRECISION SKEW
C                ! skewness
      DOUBLE PRECISION STD
C                ! standard deviation
      DOUBLE PRECISION STD2
C                ! similar to STD, differing by ((n-1)/n)**.5 []
      DOUBLE PRECISION SUMX
C                ! sum of parameter values [-]
      DOUBLE PRECISION SUMX2
C                ! sum of square of parameter values [-]
      DOUBLE PRECISION SUMX3
C                ! sum of cube of parameter values [-]
      DOUBLE PRECISION SUMX4
C                ! sum of fourth power of parameter values [-]
      DOUBLE PRECISION SUMLX
C                ! sum of ln(parameter values) [-]
      DOUBLE PRECISION SUMLX2
C                ! sum of square of ln(parameter values) [-]
      DOUBLE PRECISION VAR
C                ! variance
```

```
      COMMON /BGSAV/
     1 SUMX(250),
     1 SUMX2(250),
     1 SUMX3(250),
     1 SUMX4(250),
     1 SUMLX(250),
     1 SUMLX2(250),
     1 RUNNUM
C                ! common info generated in BGSAVE
C
      NOX  = RUNNUM
      NPAR = NSPAR+NUMCQ
      WRITE (IULPT, 300) RUNNUM
  300 FORMAT (/,/,/,/,' STATISTICAL SUMMARY  -  ',I3,' RUNS',
     1       /,1H ,'********************',/,/,/,
     1       1H ,' NO',2X,'PARAMETER',8X,'MEAN',9X,'STD',10X,
     1       'GM',11X,'GSD',10X,'SKEW',9X,'KURTOSIS')
C
      IF (RUNNUM .LE. 0) GO TO 360
C
      DO 350 I=1,NPAR
         IF (PARTYP(I) .NE. 'LIT' .AND. DSTTPS(I).NE.'CONST' ) THEN
            MEAN = SUMX(I)/NOX
            VAR  = (SUMX2(I) - NOX*MEAN**2) / (NOX - 1.0D00)
C           AVOID ATTEMPTING NEGATIVE SQUARE ROOTS, WHICH COULD
C           OCCUR FOR CONSTANT PARAMETERS
            IF (VAR .GT. 0.0D00) THEN
               STD  = SQRT (VAR)
               STD2 = SQRT ( (SUMX2(I) - NOX*MEAN**2)/ NOX )
               SKEW = ( SUMX3(I) - 3.0D00*MEAN*SUMX2(I) +
     1                  2.0D00*NOX*MEAN**3 ) / NOX / STD2**3
               KURT = ( SUMX4(I) - 4.0D00*MEAN*SUMX3(I) +
     1                  6.0D00*SUMX2(I)*MEAN**2 - 3.0D00*NOX*MEAN**4 )
     1                  / NOX / STD2**4
            ELSE
               STD  = 0.0D00
               STD2 = 0.0D00
               SKEW = 0.0D00
               KURT = 0.0D00
            ENDIF
            GM   = EXP ( SUMLX(I)/NOX )
C           AVOID ATTEMPTING NEGATIVE SQUARE ROOTS, AS ABOVE
            GSD  =    (SUMLX2(I) - SUMLX(I)**2/NOX) / (NOX - 1.0D00)
            IF (GSD .GT. 0.0D00) THEN
               GSD = EXP ( SQRT (GSD) )
            ELSE
               GDS = 0.0D00
            ENDIF
            WRITE (IULPT,310) I,PSNAME(I),MEAN,STD,GM,GSD,SKEW,KURT
  310       FORMAT (1H ,I3,2X,A14, 6(E11.4,2X) )
         ENDIF
  350 CONTINUE
  360 RETURN
      END
      SUBROUTINE BISECT (A, B, ACCUR, FUNC, ROOT, IERR)
C
C**************************************************************** BISECT
C  FINDS THE ROOT OF A FUNCTION BETWEEN TWO SPECIFIED LIMITS, USING
C  A BISECTION ALGORITHM.  BASED ON THE DESCRIPTION BY R.W. HAMMING
C  IN 'INTRODUCTION TO APPLIED NUMERICAL ANALYSIS', MCGRAW-HILL,
C  1971, CHPT. 2.
C
C  86-FEB-25  VERSION 01  B.GOODWIN
C  86-MAR-04  VERSION 02  B.GOODWIN
C  86-MAR-12  VERSION 03  B.GOODWIN
C
C***********************************************************************
C
      IMPLICIT LOGICAL (A-Z)
      DOUBLE PRECISION A, B
C                 ! limits defining a range within which a
C                   root is located [X]
      DOUBLE PRECISION ACCUR
C                 ! relative accuracy desired in the root.  IERR is set
C                   if ACCUR is less that 1.0D-12 [X/X]
      DOUBLE PRECISION FUNC
C                 ! external function to be scanned.  Calls to FUNC
C                   are of the form:  Z = FUNC (X) where X is the
C                   independent variable.  [Z]
      DOUBLE PRECISION ROOT
C                 ! contains the root found [X]
      INTEGER IERR
C                 ! error code with the following values & meaning []:
C                   IERR = 0  - no errors detected, root found
C                   IERR = 1  - interval contains no roots or an even
C                               number of roots.  None have been found;
C                               re-use bisect with a smaller range.
C                   IERR = 2  - A and B are identical and are not a
C                               root.
C                   IERR = 3  - no roots found, but a pole has
C                               has probably been found.
C                   IERR = 4  - specified accuracy is less that 1.0D-12
      INTEGER J
C                 ! loop index []
      INTEGER NOTRY
C                 ! number of bisections required to acheive the
C                   specified accuracy. []
      DOUBLE PRECISION F1, F2, F3
C                 ! temporary variables used to store values of FUNC
C                   evaluated at X1, X2 and X3.  [Z]
      DOUBLE PRECISION X1, X2, X3
C                 ! temporary variables used to store values of the
C                   independent variable required by FUNC.  [X]
      DOUBLE PRECISION ZERO
C                 ! set equal to 0.0D00 for comparisons [X]
      DATA ZERO /0.0D00/
C ......CHECK FOR 'REASONABLE' ACCURACY
      IF (ACCUR .LT. 1.0D-12) THEN
         IERR = 4
      ELSE IF ( ABS(A-B) .EQ. ZERO) THEN
C ......ARE A AND B IDENTICAL AND AT A ROOT?
         IF (FUNC(A) .EQ. ZERO) THEN
            IERR = 0
            ROOT = A
         ELSE
            IERR = 2
         ENDIF
      ELSE
C  ......SEARCH FOR A ROOT BETWEEN A AND B INCLUSIVE.
         X1 = A
         X2 = B
         F2 = FUNC(X2)
         F1 = FUNC(X1)
C  .........AT LEAST ONE EXISTS IF FUNC(A) AND FUNC(B) HAVE OPPOSITE
C           SIGNS.  IF THEY HAVE THE SAME SIGN, THERE COULD BE ZERO
C           OR AN EVEN NUMBER OF ROOTS - AN ERROR CONDITION.
         IF (F1*F2 .GT. ZERO) THEN
            IERR=1
C  .........IS THERE A ROOT AT EITHER EXTREME?
         ELSE IF (F1 .EQ. ZERO) THEN
            IERR = 0
            ROOT = X1
         ELSE IF (F2 .EQ. ZERO) THEN
            IERR = 0
            ROOT = X2
         ELSE
C  ...............EACH SEARCH WILL USE UP TO NOTRY BISECTIONS, WHICH
C                 WILL GUARANTEE THE REQUIRED ACCURACY.  THE CONSTANT
C                 BELOW IS ALOG(2)
            NOTRY = INT ( ( LOG( ABS(B-A) ) - LOG(ACCUR) )
     1                 / 0.6931471806D00 )
            DO 330 J = 1, NOTRY
               X3 = 0.5D00 * (X1 + X2)
               F3 = FUNC (X3)
               IF ( (F1 * F3) .GT. ZERO) THEN
C  ...............ROOT COULD BE BETWEEN X3 AND X2
                   X1 = X3
                   F1 = F3
               ELSE IF ( (F1 * F3) .LT. ZERO) THEN
C  ................ROOT COULD BE BETWEEN X3 AND X1
                   X2 = X3
                   F2 = F3
               ELSE
C  ...............FOUND A ROOT BANG ON; SAVE IT
                   IERR = 0
                   ROOT = X3
                   GO TO 350
               ENDIF
 330        CONTINUE
C  ............SOMETHING HAS NOW BEEN LOCATED WITHING THE DESIRED
C                  ACCURACY.  IT IS ASSUMED TO BE A ROOT IF F3 LIES
C                  BETWEEN F1 AND F2; OTHERWISE IT IS (PROBABLY) A
C                  POLE.  CALCULATE A NEW X3 AND F3 AND TEST.
            X3 = 0.5D00 * (X1 + X2)
            F3 = FUNC(X3)
            IF ( (F3-F2)*(F1-F3) .GT. ZERO) THEN
               IERR = 0
               ROOT = X3
            ELSE
               IERR = 3
            ENDIF
         ENDIF
      ENDIF
 350  CONTINUE
```

```
C         WRITE (6,3333) IERR,X1,X2,X3,F1,F2,F3
C 3333    FORMAT (1H , 'BISECT IERR=',I3,/5X,'X1-X3,F1-F3:',/6E12.3)
      RETURN
      END
```

```
C
C************************************************************************* CPALL
C
C Calculated parameters (actually calculated variables) used in
C    uranium tailings model.
C
C 86-FEB-28  VERSION 01  T. ANDRES
C 86-MAR-12  VERSION 02  B. GOODWIN
C
C***************************************************************************
C
      DOUBLE PRECISION ACT14
C                ! activity coefficient in the lake raised to
C                  the power 1/4, or ACTCO ** 1/4   []
      DOUBLE PRECISION ACT34
C                ! ACTCO ** 3/4  []
      DOUBLE PRECISION ACT94
C                ! ACTCO ** 9/4  []
      DOUBLE PRECISION AEQBM
C                ! sq. root of eqbm. const. [mol m**-3]
      DOUBLE PRECISION CRAGIN
C                ! init. conc. of Ra in gypsum[mol m**-3]
      DOUBLE PRECISION CSINIT
C                ! init. conc. of sulphate at bottom of unsat. zone
C                ! [mol m**-3]
      DOUBLE PRECISION CTH230
C                ! init. solid phase conc. of Th-230 [mol m**-3]
      DOUBLE PRECISION CTH232
C                ! init. solid phase conc. of Th-232 [mol m**-3]
      DOUBLE PRECISION DIFLEN
C                ! reciprocal of Rn diffusion path length [m**-1]
      DOUBLE PRECISION GYPBLK
C                ! init. gypsum content in tailings [mol m**-3]
      DOUBLE PRECISION KRATE
C                ! rate of oxd'n of pyrite [a**-1]
      DOUBLE PRECISION RADIUM
C                ! total init. conc. of Ra in solids [mol m**-3]
      DOUBLE PRECISION RASIMM
C                ! init. conc. of immobile Ra in tailings [mol m**-3]
      DOUBLE PRECISION SFLUX
C                ! sulphate flux from oxd'n of pyrite [mol m**-2 a**-1]
      DOUBLE PRECISION TFINAL
C                ! time to deplete gypsum in unsat. zone [a]
      DOUBLE PRECISION THINIT
C                ! init. total Th in solid phase [mol m**-3]
      DOUBLE PRECISION THINTH
C                ! mf of Th-230 to Th-232 [-]
      DOUBLE PRECISION XLAML
C                ! retention time in lake [a]
C
      COMMON /CPALL/
     1 ACT14,
     1 ACT34,
     1 ACT94,
     1 AEQBM,
     1 CRAGIN,
     1 CSINIT,
     1 CTH230,
     1 CTH232,
     1 DIFLEN,
     1 GYPBLK,
     1 KRATE,
     1 RADIUM,
     1 RASIMM,
     1 SFLUX,
     1 TFINAL,
     1 THINIT,
     1 THINTH,
     1 XLAML
C
```

```
C
C********************************************************************* CQFISH
C
C Fish Dose Consequences
C
C 86-MAR-03  VERSION 01  T. ANDRES
C
C*****************************************************************************
C
      DOUBLE PRECISION DSFISH
C                   ! selected values from FISHDS
C                   ! dose from ingestion of fish [Sv a**-1]
      DOUBLE PRECISION MAXFDS
C                   ! maximum values of FISHDS [Sv a**-1]
      DOUBLE PRECISION TMXFDS
C                   ! time of maximum FISHDS [a]
      DOUBLE PRECISION AVGFDS
C                   ! average value of FISHDS [Sv a**-1]
C
      COMMON /CQFISH/
     1 DSFISH(11),
     1 MAXFDS,
     1 TMXFDS,
     1 AVGFDS
C
```

```
C
C********************************************************************* CQLAPH
C
C Lake pH Consequences
C
C 86-MAR-04  VERSION 01  T. ANDRES
C
C****************************************************************************
C
      DOUBLE PRECISION LAKPH
C                     ! selected values from PHLAKE
C                     ! pH of lake [-]
      DOUBLE PRECISION MAXLPH
C                     ! maximum values of PHLAKE [-]
      DOUBLE PRECISION TMXLPH
C                     ! time of maximum PHLAKE [a]
      DOUBLE PRECISION AVGLPH
C                     ! average value of PHLAKE [-]
C
      COMMON /CQLAPH/
     1 LAKPH(11),
     1 MAXLPH,
     1 TMXLPH,
     1 AVGLPH
C
```

```
C
C******************************************************************** CQMEAT
C
C Caribou Meat Ingestion Consequences
C
C 86-MAR-03  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      DOUBLE PRECISION DSMEAT
C                 ! selected values from XMEATD
C                 ! dose from ingestion of caribou [Sv a**-1]
      DOUBLE PRECISION MAXMDS
C                 ! maximum values of XMEATD [Sv a**-1]
      DOUBLE PRECISION TMXMDS
C                 ! time of maximum XMEATD [a]
      DOUBLE PRECISION AVGMDS
C                 ! average value of XMEATD [Sv a**-1]
C
      COMMON /CQMEAT/
     1 DSMEAT(11),
     1 MAXMDS,
     1 TMXMDS,
     1 AVGMDS
C
```

```
C
C******************************************************************** CQRND
C
C Rn Inhalation Consequences
C
C 86-MAR-03  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      DOUBLE PRECISION DSIRN
C                 ! selected values from RNDOS
C                 ! dose from inhalation of Rn [Sv a**-1]
      DOUBLE PRECISION MAXRND
C                 ! maximum values of RNDOS [Sv a**-1]
      DOUBLE PRECISION TMXRND
C                 ! time of maximum RNDOS [a]
      DOUBLE PRECISION AVGRND
C                 ! average value of RNDOS [Sv a**-1]
C
      COMMON /CQRND/
     1 DSIRN(11),
     1 MAXRND,
     1 TMXRND,
     1 AVGRND
C
```

```
C
C********************************************************************* CQTDOS
C
C Total Dose Consequences
C
C 86-MAR-03  VERSION 01  T. ANDRES
C
C****************************************************************************
C
      DOUBLE PRECISION DSTOT
C                     ! selected values from TDOSE
C                     ! total annual dose [Sv a**-1]
      DOUBLE PRECISION MAXTOT
C                     ! maximum values of TDOSE [Sv a**-1]
      DOUBLE PRECISION TMXTOT
C                     ! time of maximum TDOSE [a]
      DOUBLE PRECISION AVGTOT
C                     ! average value of TDOSE [Sv a**-1]
C
      COMMON /CQTDOS/
     1 DSTOT(11),
     1 MAXTOT,
     1 TMXTOT,
     1 AVGTOT
C
```

```
C
C********************************************************************* CQVEG
C
C Vegetation Ingestion Consequences
C
C 86-MAR-03  VERSION 01  T. ANDRES
C
C***************************************************************************
C
      DOUBLE PRECISION DSTVEG
C                     ! selected values from TVEGDS
C                     ! dose from ingestion of vegetables [Sv a**-1]
      DOUBLE PRECISION MAXVDS
C                     ! maximum values of TVEGDS [Sv a**-1]
      DOUBLE PRECISION TMXVDS
C                     ! time of maximum TVEGDS [a]
      DOUBLE PRECISION AVGVDS
C                     ! average value of TVEGDS [Sv a**-1]
C
      COMMON /CQVEG/
     1 DSTVEG(11),
     1 MAXVDS,
     1 TMXVDS,
     1 AVGVDS
C
```

```
C
C Dust Inhalation Consequences
C
C 86-MAR-03  VERSION 01  T. ANDRES
C
      DOUBLE PRECISION DSDUST
C                 ! selected values of XINDOS
C                 ! dose from inhalation of dust [Sv a**-1]
      DOUBLE PRECISION MAXXIN
C                 ! maximum values of XINDOS [Sv a**-1]
      DOUBLE PRECISION TMXXIN
C                 ! time of maximum XINDOS [a]
      DOUBLE PRECISION AVGXIN
C                 ! average value of XINDOS [Sv a**-1]
C
      COMMON /CQXIN/
     1 DSDUST(11),
     1 MAXXIN,
     1 TMXXIN,
     1 AVGXIN
C
```

```
      DOUBLE PRECISION FUNCTION DEPFNC(H)
C
C***************************************************************DEPFNC
C
C  Returns the value of F(H) where F(H) is a modified version of
C  equation (A.3.20) in SENES Vol. 3 and in Figure 1.2.4.1.  The
C  modified equation is:  (l.h.s) - (r.h.s).
C
C  86-APR-01   VERSION 01   T. ANDRES
C
C********************************************************************
C
      IMPLICIT LOGICAL (A-Z)
C
      INCLUDE 'CPALL.INC'
C                  ! for AEQBM, CSINIT, KRATE, GYPBLK
      INCLUDE 'DICOM.INC'
C                  ! for T
      INCLUDE 'SPDEP.INC'
C                  ! for DUNSAT, NPRCP
C
      DOUBLE PRECISION EX
C                  !intermediate param. = EXP(-KRATE*T) [-]
      DOUBLE PRECISION H
C                  ! estimate of depth of upper unsat. zone [m]
      DOUBLE PRECISION ROOT1
C                  !intermediate param.  [mol/m**3]
      DOUBLE PRECISION ROOT2
C                  !intermediate param.  [mol/m**3]
      DOUBLE PRECISION S
C                  !intermediate param.  [mol/m**3]
C
      EX     = EXP( - KRATE * T )
      S      = CSINIT * H / (2.D0 * DUNSAT)
      ROOT1  = SQRT(AEQBM**2 +    S  **2)
      ROOT2  = SQRT(AEQBM**2 + (EX*S)**2)
      DEPFNC = (GYPBLK*H) - (NPRCP*AEQBM*T) -
     1         (NPRCP/KRATE) *
     1         (  (ROOT1-ROOT2)  -
     1            S * (1.0D0-EX) -
     1            AEQBM * LOG((AEQBM+ROOT1) / (AEQBM+ROOT2))
     1         )
      RETURN
      END
      SUBROUTINE DEPPAR(IULPT,IUTRM,ACCEPT)
C
C**************************************************************** DEPPAR
C
C Calculate dependant parameters, those which depend on the sampled
C parameters but are independent of time.
C
C  84-APR-03  VERSION 01  G. SHERMAN
C  86-MAR-4   VERSION 10  B. GOODWIN
C  86-MAR-30  VERSION 11  T. ANDRES
C
C***********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'TF.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*48
C                ! error message []
      DOUBLE PRECISION TFFUNC
      EXTERNAL TFFUNC
C                ! function whose root is TFINAL
      INTEGER IERR
C                ! error number returned by BISECT []
      INTEGER IULPT
C                ! FORTRAN unit number of the
C                ! print file []
      INTEGER IUTRM
C                ! FORTRAN unit number of
C                ! interactive terminal []
      LOGICAL ACCEPT
C                ! acceptance flag []
      LOGICAL STOPP
C                ! stop program flag []
      DATA
     1 MODNAM/'DEPPAR'/,
     1 MSG/'BISECT DID NOT FIND ONE VALID ROOT FOR TFINAL   '/,
     1 STOPP/.FALSE./
C
C.....assume results will be acceptable
      ACCEPT = .TRUE.
C.......................................................................
C.....INIT. GYPSUM CONTENT IN TAILINGS (GYPBLK) -           SEE FIG. 1.1
      GYPBLK = RHOTAL * FRACGP / 136.
C.......................................................................
C.....SQ. ROOT OF EQBM. CONST. (AEQBM) -                SEE FIG. 1.2.1.4
      AEQBM = SQRT(KSUBSO) / FACTCO
C.......................................................................
C.....SULPHATE FLUX FROM OXD'N OF PYRITE (SFLUX) -      SEE FIG. 1.2.1.1
      SFLUX =  SQRT ( (8.0/7.5) * SULPRD * RHOTAL * O2CONC * O2DIFF *
     1        (1.0 - KPRIME * LOG ( (1.0 + KPRIME) / KPRIME) ) )
C.......................................................................
C.....INIT. CONC OF SULPHATE AT BOTTOM OF UNSAT. ZONE (CSINIT) -
C.....                                                  SEE FIG. 1.2.1.2
```

```
      CSINIT = SFLUX / NPRCP
C........................................................................
C.....RATE OF OXD'N OF PYRITE (KRATE) -                   SEE FIG. 1.2.1.3
      KRATE = SFLUX / (2.0 * DUNSAT * PYRITE)
C........................................................................
C.....TIME TO DEPLETE GYPSUM IN UNSAT. ZONE (TFINAL) -     SEE FIG. 1.2.2
C.....CALCULATED USING THE BISECTION ALGORITHM (BISECT) WITH THE
C.....FOLLOWING CONTROL PARAMETERS:  LOWER LIMIT = 1 A; UPPER LIMIT =
C.....5000 A; AND ACCURACY = 1.0E-8.  THE REQUIRED 'SEARCH' FUNCTION
C.....IS CALLED TFFUNC, AND THE RESULT IS RETURNED DIRECTLY TO
C.....TFINAL.  NOTE:  ANY VALUE OF IERR OTHER THAN ZERO MEANS TFINAL
C.....COULD NOT BE FOUND.
      DEPTH = DUNSAT
      CALL BISECT (1.0D00, 5000.0D00, 1.0D-8, TFFUNC, TFINAL, IERR)
      IF (IERR .NE. 0) THEN
         CALL WRERR(MODNAM,MSG,STOPP)
         ACCEPT = .FALSE.
      ENDIF
C........................................................................
C.....TOTAL INIT. CONC. OF RA IN SOLIDS (RADIUM) -          SEE FIG. 1.3.1
      RADIUM = 1.1960E-13 * CONCRA * RHOTAL
C........................................................................
C.....INIT. CONC. OF RA IN GYPSUM (CRAGIN) -                SEE FIG. 1.3.2
      CRAGIN = RAINGP * RADIUM
C........................................................................
C.....INIT. SOLID PHASE CONC. OF TH-230 (CTH230) -          SEE FIG. 1.3.3
      CTH230 = RADIUM * RALAM / THLAM
C........................................................................
C.....INIT. CONC. OF IMMOBILE RA IN TAILINGS (RASIMM) -   SEE FIG. 1.3.4
      RASIMM = (1 - RAINGP) * RADIUM
C........................................................................
C.....INIT. SOLID PHASE CONC. OF TH-232 (CHT232) -      SEE FIG. 1.3.7.1
      CTH232 = UO2TON * RHOTAL * 3.788E-6 / UPERTH
C........................................................................
C.....MF OF TH-230 TO TH-232 (THINTH) -                  SEE FIGURE 1.3.7.3
      THINTH = CTH230 / CTH232
C........................................................................
C.....INIT. TOTAL TH IN SOLID PHASE (THINIT) -           SEE FIG. 1.3.7.2
      THINIT = CTH230 + CTH232
C........................................................................
C.....RETENTION TIME IN THE LAKE (XLAML) -                  SEE FIG. 4.1.3
      XLAML = VLAKE / (NPRCP * AWAT)
C........................................................................
C.....RECIPROCAL OF RN DIFFUSION PATH LENGTH (DIFLEN) -   SEE FIG. 2.2.3
      DIFLEN = SQRT ( RNLAM / ( 0.106D-04 * DEXP(-0.261*MPCH2O) ) )
C........................................................................
C.....ACTCO APPEARS IN PHFUNC WITH VARIOUS EXPONENTS; CALCULATE
C     THEM HERE.  ACTCO**.25, ACTCO**2.25 AND ACTCO**.75
      ACT14 = ACTCO**0.25D00
      ACT34 = ACT14**3
      ACT94 = ACT14**9
C........................................................................
      RETURN
      END
```

```
C
C********************************************************************* D1COM
C
C Variables to be passed from ZDEPT1 to DEPFNC without going through
C    BISECT in between.
C
C 86-MAR-18  VERSION 01  T. ANDRES
C
C***************************************************************************
C
      DOUBLE PRECISION T
C                 ! current time [a]
C
      COMMON /D1COM/
     1 T
```

```
C
C************************************************************************ FRCCOM
C
C Variables to be passed from ZFRACR to FRCFNC without going through
C    BISECT in between.
C
C 86-MAR-19  VERSION 01  T. ANDRES
C
C************************************************************************
C
      INTEGER CURIND
C                 ! current index in vectors VALS1, VALS2, etc. [-]
      DOUBLE PRECISION DELTAT
C                 ! difference between current time and previous one [a]
      DOUBLE PRECISION PRVFR
C                 ! previous value of FRACRA [-]
C
      COMMON /FRCCOM/
     1 CURIND,
     1 DELTAT,
     1 PRVFR
```

```
      DOUBLE PRECISION FUNCTION FRCFNC (FRLOG)
C
C*************************************************************** FRCFNC
C
C Returns the value of f(FR) where FR = 10**FRLOG and f(FR) is a
C    modification of the iterative equation for FRACRA in process 1.4.1.
C
C Preconditions:  In TSPACE.INC, assume that VALS1 = CONCCA,
C                    VALS2 = DEPTH1, VALS3 = REPGYP, VALS4 = RAFTH.
C                 FRLOG <= 0.
C                 Current index (CURIND in FRCCOM.INC) > 1.
C
C Postconditions: FRCFNC = [left-hand-side]  -  [right-hand-side]
C                    of equation A.3.24.
C
C 86-MAR-30  VERSION 01  T. ANDRES
C
C*********************************************************************
C
      IMPLICIT LOGICAL (A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'FRCCOM.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'TSPACE.INC'
C
      DOUBLE PRECISION DELH
C                ! incremental change in DEPTH1 from TIME to TIME + DELT
C                ! [m]
      DOUBLE PRECISION FACTOR
C                ! intermediate variable [mol/m**3]
      DOUBLE PRECISION FRLOG
C                ! independent variable (log FRACRA in this case) passed
C                ! from BISECT.  [-]
      DOUBLE PRECISION FR
C                ! 10**FRLOG [-]
C
C.....In TSPACE.INC, assume that VALS1 = CONCCA, VALS2 = DEPTH1,
C.....VALS3 = REPGYP, VALS4 = RAFTH.
C
      DELH   = VALS2(CURIND) - VALS2(CURIND-1)
      FR     = 10.D0**FRLOG
C.....The following conditional expression was put in because of log
C.....arguments going out of range at the end of the first time step.
      IF ( PRVFR .GT. 0.D0 ) THEN
         FACTOR = (  (
     1                VALS1(CURIND)*FR - VALS1(CURIND-1)*PRVFR
     1                ) / LOG(
     1                (VALS1(CURIND)*FR) / (VALS1(CURIND-1)*PRVFR)
     1             )  )
      ELSE
         FACTOR = FR * VALS1(CURIND)
      END IF
      FRCFNC = CRAGIN * DELH +
     1         NPRCP * VALS4(CURIND-1) * DELTAT -
     1         VALS3(CURIND) * FR +
     1         VALS3(CURIND-1) * PRVFR -
     1         NPRCP * DISTCO * DELTAT * FACTOR
```

```
      RETURN
      END
      SUBROUTINE INITCQ(ACCEPT)
C
C************************************************************* INITCQ
C
C Initialize consequences.
C
C 86-MAR-24  VERSION 01  T. ANDRES
C
C**********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'H:PCOUNT.INC'
      INCLUDE 'H:PARDES.INC'
      INCLUDE 'H:PARLOC.INC'
      INCLUDE 'H:PARCOM.INC'
      INCLUDE 'H:PSAMPL.INC'
C
      DOUBLE PRECISION BLANKS
C                 ! for initializing literal field []
      DOUBLE PRECISION CQFISH
C                 ! consequence common block []
      DOUBLE PRECISION CQLAPH
C                 ! consequence common block []
      DOUBLE PRECISION CQMEAT
C                 ! consequence common block []
      DOUBLE PRECISION CQRND
C                 ! consequence common block []
      DOUBLE PRECISION CQTDOS
C                 ! consequence common block []
      DOUBLE PRECISION CQVEG
C                 ! consequence common block []
      DOUBLE PRECISION CQWAT
C                 ! consequence common block []
      DOUBLE PRECISION CQXIN
C                 ! consequence common block []
      DOUBLE PRECISION CQINIT
C                 ! consequence preset value []
C
      INTEGER I
C                 ! do loop index []
      INTEGER K
C                 ! parameter pointer []
C
      LOGICAL ACCEPT
C                 ! logical indicator of
C                 ! suitable parameter selection []
C
C.....define all consequence common blocks as containing arrays
      COMMON /CQFISH/
     1 CQFISH(MXSTCM)
C
      COMMON /CQLAPH/
     1 CQLAPH(MXSTCM)
C
      COMMON /CQMEAT/
     1 CQMEAT(MXSTCM)
C
      COMMON /CQRND/
     1 CQRND(MXSTCM)
C
      COMMON /CQTDOS/
     1 CQTDOS(MXSTCM)
C
      COMMON /CQVEG/
     1 CQVEG(MXSTCM)
C
      COMMON /CQWAT/
     1 CQWAT(MXSTCM)
C
      COMMON /CQXIN/
     1 CQXIN(MXSTCM)
C
      DATA
     1 BLANKS/Z'2020202020202020'/
C
      IF (SMETHD(NUMMTH) .EQ. 'CONSEQUENCES') THEN
C........initialize common block arrays
         DO 30 I = 1,NUMCQ
            IF (ACCEPT) THEN
C..............initialize to blank or zero
               IF (PARTYP(NSPAR+I) .EQ. 'LIT') THEN
                  CQINIT = BLANKS
               ELSE
                  CQINIT = 0.D0
               END IF
            ELSE
C..............initialize to "artificial value"
               IF (PARTYP(NSPAR+I) .EQ. 'LIT') THEN
                  READ (PARART(NSPAR+I)(1:8),10) CQINIT
10                FORMAT(A8)
               ELSE
                  READ (PARART(NSPAR+I),20) CQINIT
20                FORMAT(D16.0)
               END IF
            END IF
C...........assign initial value to correct common block location
            K = PSTART(NUMMTH) + I - 1
            IF (COMNAM(COMIDX(K)) .EQ. 'CQFISH') THEN
               CQFISH(PARIDX(K)) = CQINIT
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQLAPH') THEN
               CQLAPH(PARIDX(K)) = CQINIT
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQMEAT') THEN
               CQMEAT(PARIDX(K)) = CQINIT
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQRND') THEN
               CQRND(PARIDX(K)) = CQINIT
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQTDOS') THEN
               CQTDOS(PARIDX(K)) = CQINIT
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQVEG') THEN
               CQVEG(PARIDX(K)) = CQINIT
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQWAT') THEN
               CQWAT(PARIDX(K)) = CQINIT
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQXIN') THEN
               CQXIN(PARIDX(K)) = CQINIT
            END IF
30       CONTINUE
      END IF
      RETURN
      END
```

```
C
C******************************************************************* PHCOM
C
C Variables to be passed from ZPHLAK to PHFUNC without going through
C     BISECT in between.
C
C 86-MAR-18  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      DOUBLE PRECISION CALAKE
C                 ! current conc. of Ca in lake [mol/m**3]
      DOUBLE PRECISION SULAKE
C                 ! current conc. of SO4 in lake [mol/m**3]
C
      COMMON /PHCOM/
     1 CALAKE,
     1 SULAKE
```

```
      DOUBLE PRECISION FUNCTION PHFUNC (PH)
C
C*************************************************************** PHFUNC
C
C  RETURNS THE VALUE OF F(PH) WHERE F(PH) IS A MODIFICATION OF THE
C  ITERATIVE EQUATION FOR PH IN PROCESS 4.3.1 (P. C-12 OF SENES VOL. 3.)
C  F(PH) IS EQUAL TO:  [LEFT-HAND-SIDE]  -  [RIGHT-HAND-SIDE]  OF THE
C  EQUATION.
C
C     86-FEB-25  VERSION 01  B. GOODWIN
C     86-MAR-18  VERSION 04  T. ANDRES
C     86-APR-09  VERSION 05  B. GOODWIN
C
C*********************************************************************
C
      IMPLICIT LOGICAL (A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'PHCOM.INC'
      INCLUDE 'SPDEP.INC'
C
      DOUBLE PRECISION PH
C                ! independent variable (pH in this case) passed
C                  from BISECT.  [pH units]
      DOUBLE PRECISION HACTIV
C                ! activity of the aqueous hydrogen ion  [-]
C
      HACTIV= 10.D0 ** (-PH)
C
C.....CONSTANTS GIVEN IN THE EQUATION ON P. C-12 ARE COMBINED AND/OR
C     EXPRESSED SLIGHTLY DIFFERENTLY HERE; FOR EXAMPLE 3 * 10**8.5
C     APPEARS AS 9.48683D08.  ALSO THE REQUIRED POWERS OF ACTCO ARE
C     CALCULATED IN DEPPAR AND APPEAR HERE AS ACT34 (=ACTCO**3/4) ETC.
C
C.....CALAKE AND SULAKE ARE IN [MOL/M**3], BUT MUST BE IN MOL/L FOR
C     USE HERE, SINCE THE EQUILIBRIUM CONSTANTS ETC. ARE IN MOL/L
C
      PHFUNC = (9.48683D08 * HACTIV**3 / ACT94) +
     1         (1.62566D04 * HACTIV**2 / ACTCO) +
     2         ( 1.99526D0 * HACTIV )           +
     3           1.38038D-4 + (2.0 * CALAKE*1.0D-3)    -
     4         ( SULAKE*1.0D-3 * (2.0 - ACT34*5.01187D11 * HACTIV**3)/
     5                   (1.0 + ACT34*5.01187D11 * HACTIV**3)  ) -
     6         ( 1.00000D-11 / HACTIV / ACT14)
      RETURN
      END
      SUBROUTINE SELECT(IUTRM,IULPT,ACCEPT)
C
C****************************************************************** SELECT
C
C Select parameter values.
C
C  85-FEB-13  VERSION 01  D. DONAHUE
C  85-SEP-30  VERSION 04  D. DONAHUE
C  86-APR-09  VERSION 05  T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'H:PARDES.INC'
      INCLUDE 'H:PARCOM.INC'
      INCLUDE 'H:PARLOC.INC'
      INCLUDE 'H:PSAMPL.INC'
C
      CHARACTER MODNAM*6
C                 ! module name []
      CHARACTER MSG*32
C                 ! error message []
C
      DOUBLE PRECISION PARVAL(MXPAR)
C                 ! parameter values []
      DOUBLE PRECISION SPDEP1
      DOUBLE PRECISION SPDEP2
      DOUBLE PRECISION SPRATH
      DOUBLE PRECISION SPLAKE
      DOUBLE PRECISION SPAIR1
      DOUBLE PRECISION SPAIR2
      DOUBLE PRECISION SPAIR3
      DOUBLE PRECISION SPTIME
      DOUBLE PRECISION UNUSED
C                 ! common block names []
C
      INTEGER I
C                 ! do loop index []
      INTEGER IULPT
C                 ! line printer unit number []
      INTEGER IUTRM
C                 ! interactive terminal unit number []
      INTEGER K
C                 ! do loop index []
      INTEGER M
C                 ! do loop index []
C
      LOGICAL ACCEPT
C                 ! run acceptance flag []
      LOGICAL ASSNOK
C                 ! parameter assignment ok []
      LOGICAL STOPP
C                 ! signal to stop processing []
C
C.....define all sampled parameter common blocks as containing arrays
      COMMON /SPDEP1/
     1 SPDEP1(MXSTCM)
C
```

```
       COMMON /SPDEP2/
      1 SPDEP2(MXSTCM)
C
       COMMON /SPRATH/
      1 SPRATH(MXSTCM)
C
       COMMON /SPLAKE/
      1 SPLAKE(MXSTCM)
C
       COMMON /SPAIR1/
      1 SPAIR1(MXSTCM)
C
       COMMON /SPAIR2/
      1 SPAIR2(MXSTCM)
C
       COMMON /SPAIR3/
      1 SPAIR3(MXSTCM)
C
       COMMON /SPTIME/
      1 SPTIME(MXSTCM)
C
       COMMON /UNUSED/
      1 UNUSED(MXSTCM)
C
       DATA
      1 MODNAM/'SELECT'/,
      1 MSG/'PROGRAM STOPPING FOR ABOVE ERROR /,
      1 STOPP/.TRUE./
C
C.....initialize common block arrays (sampled parameters) to 0
       DO 10 I = 1,MXSTCM
          SPDEP1(I) = 0.D0
          SPDEP2(I) = 0.D0
          SPRATH(I) = 0.D0
          SPLAKE(I) = 0.D0
          SPAIR1(I) = 0.D0
          SPAIR2(I) = 0.D0
          SPAIR3(I) = 0.D0
          SPTIME(I) = 0.D0
          UNUSED(I) = 0.D0
10     CONTINUE
       ASSNOK = .TRUE.
       DO 50 M = 1,NUMMTH
          IF ( SMETHD(M) .EQ. 'CALCULATED'
      1   .OR. SMETHD(M) .EQ. 'CONSEQUENCES') THEN
             CONTINUE
          ELSE IF ( SMETHD(M) .EQ. 'RANDOM'
      1   .OR.       SMETHD(M) .EQ. 'QUANTILE'
      1   .OR.       SMETHD(M) .EQ. 'ON FILE' ) THEN
C...........sample values for sampling method
             CALL ASSVAL(IULPT,IUTRM,M,PSTART(M),PEND(M),
      1      SMETHD(M),IUSAMP(M),SEED(M),PARVAL,ASSNOK)
             IF (ASSNOK) THEN
C..............assign sampled values to correct common block location
                DO 30 K = PSTART(M),PEND(M)
                   IF (COMNAM(COMIDX(K)) .EQ. 'SPDEP1') THEN
                      SPDEP1(PARIDX(K)) = PARVAL(K)
                   ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPDEP2') THEN
                      SPDEP2(PARIDX(K)) = PARVAL(K)
                   ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPRATH') THEN
```

```
                     SPRATH(PARIDX(K)) = PARVAL(K)
                  ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPLAKE') THEN
                     SPLAKE(PARIDX(K)) = PARVAL(K)
                  ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPAIR1') THEN
                     SPAIR1(PARIDX(K)) = PARVAL(K)
                  ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPAIR2') THEN
                     SPAIR2(PARIDX(K)) = PARVAL(K)
                  ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPAIR3') THEN
                     SPAIR3(PARIDX(K)) = PARVAL(K)
                  ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPTIME') THEN
                     SPTIME(PARIDX(K)) = PARVAL(K)
                  ELSE IF (COMNAM(COMIDX(K)) .EQ. 'UNUSED') THEN
                     UNUSED(PARIDX(K)) = PARVAL(K)
                  ELSE
                     WRITE (IULPT,20) COMNAM(COMIDX(K)),PSNAME(K)
20                   FORMAT(/,' **ERROR** (SELECT) COMMON BLOCK ',
     1               A6,' WHICH SHOULD CONTAIN PARAMETER ',
     1               A14,' IS MISSING.')
                     WRITE (IUTRM,20) COMNAM(COMIDX(K)),PSNAME(K)
                     ASSNOK = .FALSE.
                     GO TO 40
                  END IF
30             CONTINUE
            END IF
40          CONTINUE
         ELSE
            WRITE(IULPT,45)M,NUMMTH,SMETHD(M)
45          FORMAT(/,' **ERROR** (SELECT) SAMPLING METHOD ',
     1      I2,' OUT OF ',I2,' (',A12,') IS NOT RECOGNIZED.')
            WRITE(IUTRM,45)M,NUMMTH,SMETHD(M)
            ASSNOK = .FALSE.
         END IF
50    CONTINUE
      IF (ASSNOK) THEN
C........determine calculated (dependent) parameter values
         CALL DEPPAR(IULPT,IUTRM,ACCEPT)
C........initialize consequence values
         CALL INITCQ(ACCEPT)
      ELSE
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      RETURN
      END
      SUBROUTINE SIMAIR(EMRARA,EMRARN,RNDOS,XMEATD,TVEGDS,XINDOS,OK)
C
C****************************************************************** SIMAIR
C
C Simulate air path doses.
C
C  86-APR-02  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'SPAIR.INC'
      INCLUDE 'SPLAKE.INC'
C
C
      CHARACTER MODNAM*6
C                 ! module name []
      CHARACTER MSG*20
C                 ! error message []
      CHARACTER SUBNUM*3(4)
C                 ! '(i)' [-]
C
      DOUBLE PRECISION FACTOR
C                 ! scalar multiplier used in calling SCAMUL [varies]
C
      INTEGER CAIRRA
C                 ! ts for CAIR(Ra) = conc. of contaminant (Ra-226) in
C                 ! air at receptor [Bq/m**3]
      INTEGER CAIRRN
C                 ! ts for CAIR(Rn) = conc. of contaminant (Rn-222) in
C                 ! air at receptor [Bq/m**3]
      INTEGER CMEAT(2)
C                 ! concentration of contaminant (Ra-226) in caribou
C                 ! meat [Bq/g(wet weight)] (for vegetation types 1 and
C                 ! 2)
      INTEGER CVEG(4)
C                 ! ts for conc. of contaminant (Ra-226) in edible
C                 ! vegetable [Bq/g(wet weight)] (for four types of
C                 ! vegetation: lichen, grass, above- and below-ground)
      INTEGER EMRARA
C                 ! ts for EMRATE(Ra) = emission rate of Ra in dust from
C                 ! tailings surface [Bq/s]
      INTEGER EMRARN
C                 ! ts for EMRATE(Rn) = emission rate of Rn gas from
C                 ! tailings [Bq/s]
      INTEGER J
C                 ! loop counter [-]
      INTEGER NONUC
C                 ! indicator of no nuclide [-]
      INTEGER ONE
C                 ! 1 [-]
      INTEGER RNDOS
C                 ! ts for dose from inhalation of Rn [Sv/a]
      INTEGER TCMEAT
C                 ! ts for total concentration of contaminant (Ra-226)
C                 ! in caribou meat [Bq/g(wet weight)]
      INTEGER TVEGDS
C                 ! ts for dose from ingestion of vegetables [Sv/a]
      INTEGER TWO
```

```
C                ! 2 [-]
      INTEGER VEGDOS(4)
C                ! ts for dose from ingestion of vegetation type v
C                ! [Sv/a] (only v = 3 and v = 4 are used)
      INTEGER XINDOS
C                ! ts for dose from inhalation of dust [Sv/a]
      INTEGER XMEATD
C                ! ts for dose from ingestion of caribou [Sv/a]
C
      LOGICAL OK
C                ! indicator that all is well in the simulation [-]
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'SIMAIR'/,
     1 MSG/'THIS RUN PUT ON HOLD'/,
     1 NONUC/0/,
     1 ONE/1/,
     1 STOPP/.FALSE./,
     1 SUBNUM/'(1)','(2)','(3)','(4)'/
     1 TWO/2/
C
C.....initialize
C      WRITE(*,900) EMRARA,EMRARN
C900   FORMAT(' ** SIMAIR ** CALLED WITH EMRARA =',I7,', EMRARN =',I7)
C.....Calculate conc. of contaminants (Ra-226 and Rn-222) in air at
C.....receptor according to process 2.3.
      FACTOR = 2.032D0 * FWIND2 * EXP(-HSRC**2/(2.D0*VDISP**2)) /
     1         (UWIND * VDISP * XDIST)
      CALL SCAMUL(FACTOR,EMRARA,'CALCULATE CAIR(RA)',CAIRRA,OK)
      CALL SCAMUL(FACTOR,EMRARN,'CALCULATE CAIR(RN)',CAIRRN,OK)
C.....calculate dose from inhalation of Rn according to process 3.6.
      FACTOR = 1.09D-4 * F * OF
      CALL SCAMUL(FACTOR,CAIRRN,'CALCULATE RNDOS',RNDOS,OK)
C.....calculate dose from inhalation of dust according to process 3.7.
      FACTOR = DCFINH * XINHIN * OF
      CALL SCAMUL(FACTOR,CAIRRA,'CALCULATE XINDOS',XINDOS,OK)
C.....calculate conc. of contaminant (Ra-226) in edible vegetables
C.....according to process 2.4, dose from ingestion of vegetation
C.....according to process 3.4, and concentration of contaminant
C.....(Ra-226) in caribou meat from vegetation types 1 and 2 according
C.....to process 2.5
      DO 10 J = 1,4
C.....invariant: J-1 vegetable types have had CVEG evaluated.
         FACTOR = DEPVEL * FDEPSF(J) * FDEPED(J) *
     1            ( 1.D0 - EXP(-VEGLAM(J)*DUREXP(J)) ) /
     1             ( YDEN(J) * VEGLAM(J) )
         CALL SCAMUL(FACTOR,CAIRRA,'CALCULATE CVEG'//SUBNUM(J),CVEG(J),
     1      OK)
         IF ( J .LE. 2 ) THEN
            FACTOR = FIR * FBI * FAUT(J)
            CALL SCAMUL(FACTOR,CVEG(J),'CALCULATE CMEAT'//SUBNUM(J),
     1         CMEAT(J),OK)
         ELSE
            FACTOR = DCFING * VEGIN(J) * FRACV(J) * FLOC(J)
            CALL SCAMUL(FACTOR,CVEG(J),'CALCULATE VEGDOS'//SUBNUM(J),
     1         VEGDOS(J),OK)
         END IF
10    CONTINUE
```

```
C.....calculate dose from ingestion of vegetables according to process
C.....3.4.
      CALL SUM(NONUC,TWO,VEGDOS(3),'CALCULATE TVEGDS',TVEGDS,OK)
C.....calculate dose from ingestion of caribou according to processes
C.....2.5 and 3.5.
      CALL SUM(NONUC,TWO,CMEAT,'CALCULATE TOTAL CMEAT',TCMEAT,OK)
      FACTOR = DCFING * XMEATI * FMEAT
      CALL SCAMUL(FACTOR,TCMEAT,'CALCULATE XMEATD',XMEATD,OK)
C902   FORMAT(' ** SIMAIR ** RETURNING WITH        RNDOS =',I7,/,
C     1        ' XINDOS = ',I6,', TVEGDS = ',I6,', XMEATD = ',I6,/,
C     1        ' AND OK = ',L6)
      RETURN
      END
      SUBROUTINE SIMDOS(FISHDS,WATDOS,RNDOS,XMEATD,TVEGDS,XINDOS,
     1   PHLAKE,OK)
C
C*************************************************************** SIMDOS
C
C Calculate total dose and evaluate consequences.
C
C 86-MAY-01  VERSION 01  T. ANDRES
C
C**********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CQFISH.INC'
      INCLUDE 'CQLAPH.INC'
      INCLUDE 'CQMEAT.INC'
      INCLUDE 'CQRND.INC'
      INCLUDE 'CQTDOS.INC'
      INCLUDE 'CQVEG.INC'
      INCLUDE 'CQWAT.INC'
      INCLUDE 'CQXIN.INC'
      INCLUDE 'SPTIME.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER MODNAM*6
C                 ! module name []
      CHARACTER MSG*20
C                 ! error message []
C
      DOUBLE PRECISION DUMMY
C                 ! subroutine argument that is not used [-]
      DOUBLE PRECISION TIMS(11)
C                 ! times at which consequences are observed [a]
      DOUBLE PRECISION TSAREA
C                 ! area under a dose time series [Sv]
C
      INTEGER FISHDS
C                 ! ts for dose from ingestion of fish [Sv/a]
      INTEGER J
C                 ! loop index [-]
      INTEGER NONUC
C                 ! indicator that no nuclide is being used [-]
      INTEGER NT
C                 ! number of times in consequence [-]
      INTEGER NTS
C                 ! number of time series in TS [-]
      INTEGER ONE
C                 ! 1 [-]
      INTEGER PHLAKE
C                 ! ts for pH of lake [-]
      INTEGER RNDOS
C                 ! ts for dose from inhalation of Rn [Sv/a]
      INTEGER TDOSE
C                 ! ts for total annual dose [Sv/a]
      INTEGER TS(6)
C                 ! ts for doses to pass to SUM [Sv/a]
      INTEGER TVEGDS
C                 ! ts for dose from ingestion of vegetables [Sv/a]
      INTEGER WATDOS
C                 ! ts for dose from ingestion of water [Sv/a]
```

```
      INTEGER XINDOS
C                 ! ts for dose from inhalation of dust [Sv/a]
      INTEGER XMEATD
C                 ! ts for dose from ingestion of caribou [Sv/a]
C
      LOGICAL OK
C                 ! indicator that all is well in the simulation [-]
      LOGICAL STOPP
C                 ! signal to stop processing []
C
      DATA
     1 MODNAM/'SIMDOS'/,
     1 MSG/'THIS RUN PUT ON HOLD'/,
     1 NONUC/0/,
     1 ONE/1/,
     1 STOPP/.FALSE./
C
C.....initialize
      WRITE(*,900) FISHDS,WATDOS,RNDOS,XMEATD,TVEGDS,XINDOS,PHLAKE
900   FORMAT(' ** SIMDOS ** CALLED WITH FISHDS = ',I6,', WATDOS = ',I6,
     1   /,  '             RNDOS  = ',I6,', XMEATD = ',I6,', TVEGDS = ',I6,
     1   /,  '             XINDOS = ',I6,', PHLAKE = ',I6)
C.....evaluate total dose
      NTS   = 6
      TS(1) = FISHDS
      TS(2) = WATDOS
      TS(3) = RNDOS
      TS(4) = XMEATD
      TS(5) = TVEGDS
      TS(6) = XINDOS
      CALL SUM(NONUC,NTS,TS,'TOTAL ANNUAL DOSE',TDOSE,OK)
C.....set up times for consequence evaluation
      NT    = 11
      DO 10 J= 1,NT
C........invariant: J-1 times have been evaluated
         TIMS(J) = DBLE(J-1) * TSTEP
10    CONTINUE
C.....Store fish dose statistics.
      IF (OK) CALL RETTS(FISHDS,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,DSFISH,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGFDS = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXFDS,MAXFDS,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....Store ingestion of water dose statistics.
      IF (OK) CALL RETTS(WATDOS,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,DSWATR,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGWDS = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXWDS,MAXWDS,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....Store inhalation of Rn dose statistics.
      IF (OK) CALL RETTS(RNDOS,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,DSIRN,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGRND = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXRND,MAXRND,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....Store meat dose statistics.
      IF (OK) CALL RETTS(XMEATD,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,DSMEAT,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGMDS = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXMDS,MAXMDS,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....Store ingestion of vegetation dose statistics.
      IF (OK) CALL RETTS(TVEGDS,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,DSTVEG,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGVDS = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXVDS,MAXVDS,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....Store inhalation of dust dose statistics.
      IF (OK) CALL RETTS(XINDOS,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,DSDUST,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGXIN = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXXIN,MAXXIN,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....Store total annual dose statistics.
      IF (OK) CALL RETTS(TDOSE,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,DSTOT,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGTOT = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXTOT,MAXTOT,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....Store pH of lake statistics.
      IF (OK) CALL RETTS(PHLAKE,OK)
      IF (OK) CALL INTETS(NT,TIMS,ONE,LAKPH,OK)
      IF (OK) CALL AREATS(ONE,TSAREA,DUMMY,DUMMY,DUMMY,OK)
      AVGLPH = TSAREA / TLIMIT
      IF (OK) CALL MAXTTS(TLIMIT,TMXLPH,MAXLPH,OK)
      IF (OK) CALL DROPTS(ONE,OK)
C.....final stub output
C      WRITE(*,902) OK
C902   FORMAT(' ** SIMDOS ** RETURNING WITH OK = ',L6)
      RETURN
      END
      SUBROUTINE SIMLAT(ONHOLD)
C
C************************************************************** SIMLAT
C
C Simulate system.
C
C  85-JAN-31  VERSION 01  D. DONAHUE
C  85-NOV-21  VERSION 08  D. DONAHUE
C  86-APR-02  VERSION 09  T. ANDRES
C
C*********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'H:OUTOPT.INC'
      INCLUDE 'H:TIMCPU.INC'
C
      CHARACTER MODNAM*6
C                 ! module name []
      CHARACTER MSG*20
C                 ! error message []
C
      INTEGER CONCCA
C                 ! ts for conc. of Ca aq. in the unsat zone [mol/m**3]
      INTEGER CRAAVG
C                 ! ts for avg. conc. of Ra to mixing zone [mol/m**3]
      INTEGER DEPTH1
C                 ! ts for depth of upper unsat. zone [m]
      INTEGER EMRARA
C                 ! ts for emission rate of Ra in dust from tailings
C                 ! surface [Bq/s]
      INTEGER EMRARN
C                 ! ts for emission rate of Rn gas from tailings [Bq/s]
      INTEGER FISHDS
C                 ! ts for dose from ingestion of fish [Sv/a]
      INTEGER PHLAKE
C                 ! ts for pH of lake [-]
      INTEGER REPRA
C                 ! ts for mass of Ra in reprec. gypsum [mol/m**2]
      INTEGER RNDOS
C                 ! ts for dose from inhalation of Rn [Sv/a]
      INTEGER TVEGDS
C                 ! ts for dose from ingestion of vegetables [Sv/a]
      INTEGER WATDOS
C                 ! ts for dose from ingestion of water [Sv/a]
      INTEGER XINDOS
C                 ! ts for dose from inhalation of dust [Sv/a]
      INTEGER XMEATD
C                 ! ts for dose from ingestion of caribou [Sv/a]
C
      LOGICAL OK
C                 ! indicator that all is well in the simulation [-]
      LOGICAL ONHOLD
C                 ! signal that an error has been
C                 ! encountered in the current run []
      LOGICAL STOPP
C                 ! signal to stop processing []
C
      DATA
```

```
     1 MODNAM/'SIMLAT'/,
     1 MSG/'THIS RUN PUT ON HOLD'/,
     1 STOPP/.FALSE./
C
C.....initialize
C      WRITE(*,900)
C900   FORMAT(' ** SIMLAT ** CALLED.')
      CALL ZAPTSS
C.....start tracing and direct output to screen, if requested.
      CALL STRACE(INDSUB,IUSUB)
C.....call SIMMIX to simulate mixing zone and lake concentrations
      CALL SIMMIX(CONCCA,PHLAKE,OK)
C.....call SIMUNZ to simulate the unsaturated zone
      IF (OK) CALL SIMUNZ(CONCCA,CRAAVG,REPRA,DEPTH1,OK)
C.....call SIMRAN to simuate Ra, Rn emission
      IF (OK) CALL SIMRAN(REPRA,DEPTH1,EMRARA,EMRARN,OK)
C.....call SIMWAT to simulate water path doses
      IF (OK) CALL SIMWAT(CRAAVG,FISHDS,WATDOS,OK)
C.....call SIMAIR to simulate air path doses
      IF (OK) CALL SIMAIR(EMRARA,EMRARN,RNDOS,XMEATD,TVEGDS,XINDOS,OK)
C.....call SIMDOS to calculate total dose and store consequences
      IF (OK) CALL SIMDOS(FISHDS,WATDOS,RNDOS,XMEATD,TVEGDS,XINDOS,
     1   PHLAKE,OK)
C.....check if the run should be put on hold
      ONHOLD = .NOT. OK
C      WRITE(*,902) ONHOLD
C902   FORMAT(' ** SIMLAT ** RETURNING WITH ONHOLD = ',L4)
      RETURN
      END
      SUBROUTINE SIMMIX(CONCCA,PHLAKE,OK)
C
C*************************************************************** SIMMIX
C
C Simulate mixing zone and lake concentrations.
C
C  86-MAY-01  VERSION 01  T. ANDRES
C
C**********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      CHARACTER MODNAM*6
C                   ! module name []
      CHARACTER MSG*20
C                   ! error message []
C
      EXTERNAL ERRABS
C                   ! add absolute errors in two-norm [-]
      EXTERNAL ERRFRC
C                   ! add fractional errors in two-norm [-]
      EXTERNAL ERRSAM
C                   ! keep error as calculated in function fitting [-]
      EXTERNAL ZCONCS
C                   ! routine to calculate CONCSB [-]
      EXTERNAL ZION
C                   ! routine to calculate ION [-]
      EXTERNAL ZCONCC
C                   ! routine to calculate CONCCA [-]
      EXTERNAL ZGYPLF
C                   ! routine to calculate GYPLFT [-]
      EXTERNAL ZSO4MI
C                   ! routine to calculate SO4MIX [-]
      EXTERNAL ZCAMIX
C                   ! routine to calculate CAMIX [-]
      EXTERNAL ZSULAK
C                   ! routine to calculate SULAKE [-]
      EXTERNAL ZCALAK
C                   ! routine to calculate CALAKE [-]
      EXTERNAL ZPHLAK
C                   ! routine to calculate PHLAKE [-]
C
      INTEGER CALAKE
C                   ! ts for conc. of Ca in lake [mol/m**3]
      INTEGER CAMIX
C                   ! ts for conc. of Ca aq in the mixing zone [mol/m**3]
      INTEGER CONCCA
C                   ! ts for conc. of Ca aq. in the unsat zone [mol/m**3]
      INTEGER CONCSB
C                   ! ts for conc. of SO4 aq in the unsat zone [mol/m**3]
      INTEGER GYPLFT
C                   ! ts for mass of gypsum left in the mixing zone
C                   ! [mol/m**2]
      INTEGER ION
C                   ! ts for conc. of ions from gyp. diss'n in mixing zone
C                   ! [mol/m**3]
      INTEGER PHLAKE
C                   ! ts for pH of lake [-]
      INTEGER SO4MIX
C                   ! ts for conc. of SO4 aq in the mixing zone [mol/m**2]
```

```
      INTEGER SULAKE
C               ! ts for conc. of S04 in lake [mol/m**3]
      INTEGER TS(5)
C               ! vector of time series indices for calls to CONVRT
C
      LOGICAL OK
C               ! indicator that all is well in the simulation [-]
      LOGICAL STOPP
C               ! signal to stop processing []
C
      DATA
     1 MODNAM/'SIMMIX'/,
     1 MSG/'THIS RUN PUT ON HOLD'/,
     1 STOPP/.FALSE./
C
C.....initialize
C      WRITE(*,900)
C900   FORMAT(' ** SIMMIX ** CALLED.')
C.....calculate conc. of S04 aq in the unsat. zone
      CALL SOURCE(0,ZCONCS,'CALCULATE CONCS8',CONCS8,OK)
C.....calculate conc. of ions from gyp. diss'n in mixing zone
      TS(1) = CONCS8
      IF (OK) CALL CONVRT(0,1,TS,ZION,'CALCULATE ION',ERRABS,ION,OK)
C.....calculate conc. of Ca aq. in the unsat. zone
      IF (OK) CALL CONVRT(0,1,TS,ZCONCC,'CALCULATE CONCCA',ERRSAM,
     1   CONCCA,OK)
C.....calculate mass of gypsum left in the mixing zone
      TS(1) = ION
      IF (OK) CALL CONVRT(0,1,TS,ZGYPLF,'CALCULATE GYPLFT',ERRSAM,
     1   GYPLFT,OK)
C.....calculate conc. of S04 and Ca aq.in the mixing zone
      TS(1) = CONCS8
      TS(2) = ION
      TS(3) = GYPLFT
      TS(4) = CONCCA
      IF (OK) CALL CONVRT(0,4,TS,ZS04MI,'CALCULATE S04MIX',ERRSAM,
     1   S04MIX,OK)
      TS(1) = CONCCA
      IF (OK) CALL CONVRT(0,3,TS,ZCAMIX,'CALCULATE CAMIX ',ERRSAM,
     1   CAMIX,OK)
C.....calculate conc. of S04 in lake
      TS(1) = S04MIX
      IF (OK) CALL CONVRT(0,1,TS,ZSULAK,'CALCULATE SULAKE',ERRFRC,
     1   SULAKE,OK)
C.....calculate conc. of Ca in lake
      TS(1) = CAMIX
      IF (OK) CALL CONVRT(0,1,TS,ZCALAK,'CALCULATE CALAKE',ERRFRC,
     1   CALAKE,OK)
C.....calculate pH of lake
      TS(1) = SULAKE
      TS(2) = CALAKE
      IF (OK) CALL CONVRT(0,2,TS,ZPHLAK,'CALCULATE PHLAKE',ERRFRC,
     1   PHLAKE,OK)
C      WRITE(*,902) CONCCA,PHLAKE,OK
C902   FORMAT(' ** SIMMIX ** RETURNING WITH',/,
C     1   ' CONCCA = ',I6,', PHLAKE = ',I6,' AND OK = ',L6)
      RETURN
      END
      SUBROUTINE SIMRAN(REPRA,DEPTH1,EMRARA,EMRARN,OK)
C
C***************************************************************** SIMRAN
C
C Simulate radon and radium emission rates.
C
C  86-APR-02  VERSION 01  T. ANDRES
C
C***********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*20
C                ! error message []
C
      EXTERNAL ERRABS
C                ! add absolute errors in two-norm [-]
      EXTERNAL ERRFRC
C                ! add fractional errors in two-norm [-]
      EXTERNAL ERRSAM
C                ! keep error as calculated in function fitting [-]
      EXTERNAL ZEMRAA
C                ! routine to calculate EMRARA [-]
      EXTERNAL ZEMRAN
C                ! routine to calculate EMRARN [-]
      EXTERNAL ZRACO1
C                ! routine to calculate RACON1 [-]
C
      INTEGER DEPTH1
C                ! ts for depth of upper unsat. zone [m]
      INTEGER NONUC
C                ! indicator of no nuclide [-]
      INTEGER REPRA
C                ! ts for mass of Ra in reprec. gypsum [mol/m**2]
      INTEGER EMRARA
C                ! ts for EMRATE(RA), emission rate of Ra in dust from
C                ! tailings surface [Bq/s]
      INTEGER EMRARN
C                ! ts for EMRATE(RN), emission rate of Rn gas from
C                ! tailings [Bq/s]
      INTEGER ONE
C                ! 1 [-]
      INTEGER RACON1
C                ! ts for MOLTOB*RACON1, Ra conc. in upper unsat. zone
C                ! [Bq/m**3]
      INTEGER THREE
C                ! 3 [-]
      INTEGER TS(5)
C                ! vector of time series indices for calls to CONVRT
C
      LOGICAL OK
C                ! indicator that all is well in the simulation [-]
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'SIMRAN'/,
     1 MSG/'THIS RUN PUT ON HOLD'/,
     1 NONUC/0/,
     1 ONE/1/,
     1 STOPP/.FALSE./,
     1 THREE/3/
C
C.....initialize
C      WRITE(*,900) DEPTH1,REPRA
C900   FORMAT(' ** SIMRAN ** CALLED WITH DEPTH1 =',I7,', REPRA =',I7)
C.....calculate Ra conc. in upper unsat. zone
      CALL SOURCE(NONUC,ZRACO1,'CALCULATE RACON1',RACON1,OK)
C.....calculate emission rate of Ra in dust from tailings surface.
      TS(1) = RACON1
      CALL CONVRT(NONUC,ONE,TS,ZEMRAA,'CALCULATE EMRATE(RA)',ERRSAM,
     1   EMRARA,OK)
C.....calculate emission rate of Rn gas from tailings [Bq/s]
      TS(1) = DEPTH1
      TS(2) = REPRA
      TS(3) = RACON1
      CALL CONVRT(NONUC,THREE,TS,ZEMRAN,'CALCULATE EMRATE(RN)',ERRFRC,
     1   EMRARN,OK)
C      WRITE(*,902) EMRARA,EMRARN,OK
C902   FORMAT(' ** SIMRAN ** RETURNING WITH',/,
C     1   ' EMRARA = ',I6,', EMRARN = ',I6,' AND OK = ',L6)
      RETURN
      END
      SUBROUTINE SIMUNZ(CONCCA,CRAAVG,REPRA,DEPTH1,OK)
C
C*********************************************************************** SIMUNZ
C
C Simulate unsaturated zone.
C
C  86-APR-02  VERSION 01  T. ANDRES
C
C*******************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      CHARACTER MODNAM*6
                 ! module name []
C
      CHARACTER MSG*20
                 ! error message []
C
C
      EXTERNAL ERRABS
                 ! routine to add absolute errors in 2-norm [-]
C
      EXTERNAL ERRFRC
                 ! routine to add fractional errors in 2-norm [-]
C
      EXTERNAL ZCRAAV
                 ! routine to calculate CRAAVG [-]
C
      EXTERNAL ZDEPT1
                 ! routine to calculate DEPTH1 [-]
C
      EXTERNAL ZFRACR
                 ! routine to calculate FRACRA [-]
C
      EXTERNAL ZRAFTH
                 ! routine to calculate RAFTH [-]
C
      EXTERNAL ZREPGY
                 ! routine to calculate REPGYP [-]
C
C
      INTEGER CONCCA
                 ! ts for conc. of Ca aq. in the unsat zone [mol/m**3]
C
      INTEGER CRAAVG
                 ! ts for avg. conc. of Ra to mixing zone [mol/m**3]
C
      INTEGER DEPTH1
                 ! ts for depth of upper unsat. zone [m]
C
      INTEGER FRACRA
                 ! ts for frac. of Ra-226 in reprecip. gypsum [-]
C
      INTEGER NONUC
                 ! nuclide number when there is none (=0) [-]
C
      INTEGER RAFTH
                 ! ts for conc. of Ra aq from Th decay in unsat. zone
C                ! [mol/m**3]
C
      INTEGER REPGYP
                 ! ts for reprec. gypsum mass in lower unsat. zone
C                ! [mol/m**2]
C
      INTEGER REPRA
                 ! ts for mass of Ra in reprec. gypsum [mol/m**2]
C
      INTEGER TS(5)
                 ! vector of time series indices for calls to CONVRT
C
C
      LOGICAL OK
                 ! indicator that all is well in the simulation [-]
C
      LOGICAL STOPP
                 ! signal to stop processing []
C
C
      DATA
     1 MODNAM/'SIMUNZ'/,
     1 MSG/'THIS RUN PUT ON HOLD'/,
     1 NONUC/0/,
     1 STOPP/.FALSE./
C
C.....initialize
C      WRITE(*,900) CONCCA
C900   FORMAT(' ** SIMUNZ ** CALLED WITH CONCCA = ',I6)
C.....calculate depth of upper unsat. zone
      CALL SOURCE(NONUC,ZDEPT1,'CALCULATE DEPTH1',DEPTH1,OK)
C.....calculate reprec. gypsum mass in lower unsat. zone
      TS(1) = DEPTH1
      CALL CONVRT(NONUC,1,TS,ZREPGY,'CALCULATE REPGYP',ERRABS,REPGYP,OK)
C.....calculate conc. of Ra aq. from Th decay in unsat. zone
      CALL SOURCE(NONUC,ZRAFTH,'CALCULATE RAFTH ',RAFTH, OK)
C.....calculate frac. of Ra in reprecip. gypsum [-]
      TS(1) = CONCCA
      TS(2) = DEPTH1
      TS(3) = REPGYP
      TS(4) = RAFTH
      CALL CONVRT(NONUC,4,TS,ZFRACR,'CALCULATE FRACRA',ERRFRC,FRACRA,OK)
C.....calculate mass of Ra in reprec. gypsum
      TS(1) = REPGYP
      TS(2) = FRACRA
      CALL MULT(NONUC,2,TS,'CALCULATE REPRA',REPRA,OK)
C.....calculate avg. conc. of Ra to mixing zone
      TS(1) = CONCCA
      TS(2) = FRACRA
      TS(3) = RAFTH
      CALL CONVRT(NONUC,3,TS,ZCRAAV,'CALCULATE CRAAVG',ERRFRC,CRAAVG,OK)
C      WRITE(*,902) CRAAVG,REPRA,DEPTH1,OK
C902   FORMAT(' ** SIMUNZ ** RETURNING WITH   CRAAVG = ',I6,/,
C     1       ' REPRA  = ',I6,', DEPTH1 = ',I6,', OK = ',L6)
      RETURN
      END
      SUBROUTINE SIMWAT(CRAAVG,FISHDS,WATDOS,OK)
C
C******************************************************************* SIMWAT
C
C Simulate water path doses.
C
C 86-APR-02  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'SPLAKE.INC'
C
C
      CHARACTER MODNAM*6
C                 ! module name []
      CHARACTER MSG*20
C                 ! error message []
C
      DOUBLE PRECISION FACTOR
C                 ! scaling factor used in calls to SCAMUL [varies]
C
      EXTERNAL ERRFRC
C                 ! routine to add fractional errors [-]
      EXTERNAL ZCLAKE
C                 ! routine to calculate CLAKE [-]
      EXTERNAL ZRAMIX
C                 ! routine to calculate MRAMIX [-]
C
      INTEGER CLAKE
C                 ! ts for Ra conc. in lake [mol/m**3]
      INTEGER CRAAVG
C                 ! ts for avg. conc. of Ra to mixing zone [mol/m**3]
      INTEGER FISHDS
C                 ! ts for dose from ingestion of fish [Sv/a]
      INTEGER MRAMIX
C                 ! ts for MOLTOB*RAMIX = conc. of Ra aq in mixing zone
C                 ! [Bq/m**3]
      INTEGER NONUC
C                 ! indicator that no nuclides are used [0]
      INTEGER ONE
C                 ! 1 [-]
      INTEGER TS(5)
C                 ! ts indices for calls to CONVRT [-]
      INTEGER WATDOS
C                 ! ts for dose from ingestion of water [Sv/a]
C
      LOGICAL OK
C                 ! indicator that all is well in the simulation [-]
      LOGICAL STOPP
C                 ! signal to stop processing []
C
      DATA
     1 MODNAM/'SIMWAT'/,
     1 MSG/'THIS RUN PUT ON HOLD'/,
     1 ONE/1/,
     1 STOPP/.FALSE./
C
C.....initialize
```

```
C      WRITE(*,900) CRAAVG
C900   FORMAT(' ** SIMWAT ** CALLED WITH CRAAVG = ',I6)
C.....calculate conc. of Ra aq in mixing zone [mol/m**3]
      TS(1) = CRAAVG
      CALL CONVRT(NONUC,ONE,TS,ZRAMIX,'CALCULATE MOLTOB*RAMIX',ERRFRC,
     1   MRAMIX,OK)
C.....calculate Ra conc. in lake
      TS(1) = MRAMIX
      IF (OK) CALL CONVRT(NONUC,ONE,TS,ZCLAKE,'CALCULATE CLAKE',ERRFRC,
     1   CLAKE,OK)
C.....calculate dose from ingestion of fish according to processes 4.2
C.....and 3.2.
      FACTOR = BF
      FACTOR = FACTOR * DCFING * FISHIN * FFISH
      IF (OK) CALL SCAMUL(FACTOR,CLAKE,'CALCULATE FISHDS',FISHDS,OK)
C.....calculate dose from ingestion of water according to process 3.3.
      FACTOR = DCFING * WATIN * FWAT
      IF (OK) CALL SCAMUL(FACTOR,CLAKE,'CALCULATE WATDOS',WATDOS,OK)
C
C      WRITE(*,902) FISHDS,WATDOS,OK
C902   FORMAT(' ** SIMWAT ** RETURNING WITH ',/,
C     1       '                   FISHDS = ',I6,', WATDOS = ',I6,', OK = ',L6)
      RETURN
      END
```

```
C
C*************************************************************** SPAIR
C
C
C Selected parameters - those needed for calculations of air-pathway
C    doses in routines called by SIMAIR.
C
C  86-MAR-10 VERSION 01  T. ANDRES
C
C*********************************************************************
C
      DOUBLE PRECISION DCFINH
C                ! whole body DCF for inhal. of Ra-226 [Sv Bq**-1]
      DOUBLE PRECISION DEPVEL
C                ! deposition velocity [m s**-1]
      DOUBLE PRECISION DUREXP
C                ! duration of exposure of vegetation [s]
      DOUBLE PRECISION F
C                ! equilibrium factor for Rn + daughters [-]
      DOUBLE PRECISION FAUT
C                ! fraction of annual intake of vegetation type v [-]
      DOUBLE PRECISION FBI
C                ! feed to meat transfer coefficient for Ra-226
C                ! [Bq g**-1 Bq**-1 a]
      DOUBLE PRECISION FDEPED
C                ! fraction on edible portion [-]
      DOUBLE PRECISION FDEPSF
C                ! fraction of deposition retained [-]
      DOUBLE PRECISION FIR
C                ! feed ingestion rate [g a**-1]
      DOUBLE PRECISION FLOC
C                ! frac. of veg. type v grown locally [-]
      DOUBLE PRECISION FMEAT
C                ! frac. of annual intake of local caribou [-]
      DOUBLE PRECISION FRACV
C                ! frac. of Ra-226 in veg. v after preparation [-]
      DOUBLE PRECISION FWIND2
C                ! fraction of time wind blows to receptor [-]
      DOUBLE PRECISION HSRC
C                ! height of source [m]
      DOUBLE PRECISION OF
C                ! frac. of year at location [-]
      DOUBLE PRECISION UWIND
C                ! mean wind speed, source to receptor [m s**-1]
      DOUBLE PRECISION VDISP
C                ! vertical dispersion coefficient [m]
      DOUBLE PRECISION VEGIN
C                ! average annual intake of veg.v (above-ground)
C                ! [g(wet weight) a**-1]
      DOUBLE PRECISION VEGLAM
C                ! weathering loss decay rate [s**-1]
      DOUBLE PRECISION XDIST
C                ! distance from source to receptor [m]
      DOUBLE PRECISION XINHIN
C                ! average annual inhalation rate [m**3 a**-1]
      DOUBLE PRECISION XMEATI
C                ! average annual intake of caribou meat [g a**-1]
      DOUBLE PRECISION YDEN
C                ! yield density [g (wet weight) m**-2]
C
```

```
      COMMON /SPAIR1/
     1 DCFINH,
     1 DEPVEL,
     1 DUREXP(4),
     1 F,
     1 FAUT(2),
     1 FBI,
     1 FDEPED(4),
     1 FDEPSF(4),
     1 FIR
C
      COMMON /SPAIR2/
     1 FLOC(4),
     1 FMEAT,
     1 FRACV(4),
     1 FWIND2,
     1 HSRC,
     1 OF,
     1 UWIND,
     1 VDISP,
     1 VEGIN(4)
C
      COMMON /SPAIR3/
     1 VEGLAM(4),
     1 XDIST,
     1 XINHIN,
     1 XMEATI,
     1 YDEN(4)
C
```

```
C
C******************************************************************* SPDEP
C
C Selected parameters - those needed for calculations of time-
C    independent calculated variables in DEPPAR, and those used in
C    routines called by SIMMIX.
C
C  86-FEB-28  VERSION 01  T. ANDRES
C
C************************************************************************
C
      DOUBLE PRECISION ACTCO
C                ! activity coefficient in lake [-]
      DOUBLE PRECISION AWAT
C                ! watershed area [m**2]
      DOUBLE PRECISION CONCRA
C                ! init. solid phase act. of Ra-226 in tailings [Bq/g]
      DOUBLE PRECISION DELT
C                ! time step-size [a]
      DOUBLE PRECISION DMIX
C                ! depth of mixing zone [m]
      DOUBLE PRECISION DUNSAT
C                ! depth of unsaturated zone [m]
      DOUBLE PRECISION FACTCO
C                ! activity coefficients for divalent ions [-]
      DOUBLE PRECISION FRACGP
C                ! mass frac. of gypsum in tailings [-]
      DOUBLE PRECISION KPRIME
C                ! ratio of KO2 to O2 [-]
      DOUBLE PRECISION KSUBSO
C                ! solubility eqm. const. for gypsum [mol**2 m**-6]
      DOUBLE PRECISION MPCH2O
C                ! percent moisture content of tailings [%]
      DOUBLE PRECISION NPRCP
C                ! net precipitation [m a**-1]
      DOUBLE PRECISION O2CONC
C                ! O2 conc. in air [mol m**-3]
      DOUBLE PRECISION O2DIFF
C                ! O2 diffusivity in tailings [m**2 a**-1]
      DOUBLE PRECISION PYRITE
C                ! init. conc. of pyrite in tailings [mol m**-3]
      DOUBLE PRECISION RAINGP
C                ! frac. of total Ra in gypsum [-]
      DOUBLE PRECISION RALAM
C                ! Ra-226 decay constant [a**-1]
      DOUBLE PRECISION RHOTAL
C                ! tailings bulk density [g m**-3]
      DOUBLE PRECISION RNLAM
C                ! Rn-222 decay constant [s**-1]
      DOUBLE PRECISION SULPRD
C                ! sulphate production rate [mol g**-1 a**-1]
      DOUBLE PRECISION THLAM
C                ! Th-230 decay constant [a**-1]
      DOUBLE PRECISION UO2TON
C                ! init. mf of UO2 in ore [kg t**-1]
      DOUBLE PRECISION UPERTH
C                ! mass ratio of UO2 to ThO2 [-]
      DOUBLE PRECISION VLAKE
C                ! lake volume [m**3]
C
```

```
 COMMON /SPDEP1/
1 ACTCO,
1 AWAT,
1 CONCRA,
1 DELT,
1 DMIX,
1 DUNSAT,
1 FACTCO,
1 FRACGP,
1 KPRIME,
1 KSUBSO,
1 MPCH2O,
1 NPRCP,
1 O2CONC,
1 O2DIFF,
1 PYRITE,
1 RAINGP,
1 RALAM,
1 RHOTAL,
1 RNLAM,
1 SULPRD
 COMMON /SPDEP2/
1 THLAM,
1 UO2TON,
1 UPERTH,
1 VLAKE
```

```
C
C************************************************************* SPLAKE
C
C Selected parameters - those needed for calculations of water-pathway
C    doses in routines called by SIMWAT.
C
C  86-MAR-01  VERSION 01  T. ANDRES
C
C*******************************************************************
C
      DOUBLE PRECISION BETA
C                ! ratio of gain rate to loss rate [-]
      DOUBLE PRECISION BF
C                ! water to fish transfer coeff. for Ra-226 [Bq g**-1
C                ! (wet weight) Bq**-1 m**3]
      DOUBLE PRECISION DCFING
C                ! whole body DCF for ingest. of Ra-226 [Sv Bq**-1]
      DOUBLE PRECISION FFISH
C                ! frac. of annual intake of local fish [-]
      DOUBLE PRECISION FISHIN
C                ! average annual intake of fish [g a**-1]
      DOUBLE PRECISION FWAT
C                ! frac. of water intake that is local [-]
      DOUBLE PRECISION KSUBDR
C                ! Ra aq-solid transfer coefficient [m**3 g**-1]
      DOUBLE PRECISION TVOID
C                ! pore volume fraction [m**3 m**-3]
      DOUBLE PRECISION WATIN
C                ! average annual intake of water [m**3 a**-1]
      DOUBLE PRECISION XK1
C                ! rate of Ra loss to sediment [a**-1]
      DOUBLE PRECISION XK3
C                ! rate of Ra burial in deep sediment [a**-1]
C
      COMMON /SPLAKE/
     1 BETA,
     1 BF,
     1 DCFING,
     1 FFISH,
     1 FISHIN,
     1 FWAT,
     1 KSUBDR,
     1 TVOID,
     1 WATIN,
     1 XK1,
     1 XK3
```

```
C
C******************************************************************* SPRATH
C
C Selected parameters - those that are related to radium and thorium
C    and those for use in routines called by SIMUNZ and SIMRAN.
C
C  86-MAR-1  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      DOUBLE PRECISION AEXPT
C                     ! exposed tailings surface area [m**2]
C                     ! same as ATAIL
      DOUBLE PRECISION ATAIL
C                     ! exposed tailings surface area [m**2]
      DOUBLE PRECISION DISTCO
C                     ! Ra solid/aq dist. coefficient [-]
      DOUBLE PRECISION EMANAT
C                     ! emanation coefficient [-]
      DOUBLE PRECISION FPREC
C                     ! annual days with sig. precip. [d]
      DOUBLE PRECISION FWIND1
C                     ! percent of time with high wind speeds [%]
      DOUBLE PRECISION MOLTOB
C                     ! conversion factor mol to Bq (Ra-226) [Bq mol**-1]
      DOUBLE PRECISION RAINTH
C                     ! frac. of ingrown Ra that is insol. [-]
      DOUBLE PRECISION SILT
C                     ! silt content of tailings surface [%]
      DOUBLE PRECISION THDISS
C                     ! rate of amorphous Th dissolution [a**-1]
      DOUBLE PRECISION THFRAC
C                     ! frac. of immobile Th in tailings [-]
C
      COMMON /SPRATH/
     1 AEXPT,
     1 ATAIL,
     1 DISTCO,
     1 EMANAT,
     1 FPREC,
     1 FWIND1,
     1 MOLTOB,
     1 RAINTH,
     1 SILT,
     1 THDISS,
     1 THFRAC
```

```
C
C*********************************************************************** SPTIME
C
C Selected parameters - times needed for storing consequence values.
C
C  86-MAY-01  VERSION 01  T. ANDRES
C
C*****************************************************************************
C
      DOUBLE PRECISION TSTEP
C                ! time step for consequence evaluations [a]
C
      COMMON /SPTIME/
     1 TSTEP
```

```
C
C****************************************************************** TF
C
C  COMMON block to pass DEPTH to TFFUNC.
C
C  86-MAR-31  VERSION 01  T. ANDRES
C
C*********************************************************************
C
      DOUBLE PRECISION DEPTH
C                     ! depth of upper layer of unsaturated zone [m]
C
      COMMON /TF/
     1 DEPTH
```

```
      DOUBLE PRECISION FUNCTION TFFUNC(T)
C
C***************************************************************TFFUNC
C
C  Returns the value of F(T) where F(T) is a modified version of
C  equation (A.3.21) in SENES Vol. 3 and in Figure 1.2.2.  The
C  modified equation is:  (r.h.s) - (l.h.s).
C
C  86-MAR-04   VERSION 01   B.GOODWIN
C  86-MAR-12   VERSION 02   B.GOODWIN
C  86-MAR-18   VERSION 02   T. ANDRES
C
C*********************************************************************
C
      IMPLICIT LOGICAL (A-Z)
C
      INCLUDE 'CPALL.INC'
C                   ! for AEQBM, CSINIT, KRATE, GYPBLK
      INCLUDE 'SPDEP.INC'
C                   ! for DUNSAT, NPRCP
      INCLUDE 'TF.INC'
C                   ! for DEPTH
C
      DOUBLE PRECISION EX
C                   !intermediate param. = EXP(-KRATE*T)
      DOUBLE PRECISION ROOT1
C                   !intermediate param.  [mol/m**3]
      DOUBLE PRECISION ROOT2
C                   !intermediate param.  [mol/m**3]
      DOUBLE PRECISION S
C                   !intermediate param.  [mol/m**3]
      DOUBLE PRECISION T
C                   !time value passed as a guess [a]
C
      EX     = EXP( - KRATE * T )
      S      = CSINIT / 2.D0
      ROOT1  = SQRT(AEQBM**2 +     S  **2)
      ROOT2  = SQRT(AEQBM**2 + (EX*S)**2)
      TFFUNC = (GYPBLK*DEPTH) - (NPRCP*AEQBM*T) -
     1         (NPRCP/KRATE) * (
     1           (ROOT1-ROOT2)  -
     1           S * (1.0D0-EX) -
     1           AEQBM * LOG((AEQBM+ROOT1) / (AEQBM+ROOT2)) )
      RETURN
      END
      SUBROUTINE WRDATA(ACCEPT,ONHOLD,RUNNUM,NACCPT,NREJCT,NHOLD)
C
C*************************************************************** WRDATA
C
C Write run data to appropriate output file(s).
C
C  83-SEP-21  VERSION 01  S. KING
C  85-OCT-02  VERSION 06  D. DONAHUE
C  86-APR-09  VERSION 07  T. ANDRES
C
C**********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'H:OUTLOC.INC'
      INCLUDE 'H:OUTOPT.INC'
      INCLUDE 'H:OUTREC.INC'
      INCLUDE 'H:PCOUNT.INC'
      INCLUDE 'H:PDISTN.INC'
      INCLUDE 'H:NUCLID.INC'
      INCLUDE 'H:PARDES.INC'
      INCLUDE 'H:PARLOC.INC'
      INCLUDE 'H:PARCOM.INC'
      INCLUDE 'H:PSAMPL.INC'
      INCLUDE 'H:TIMCPU.INC'
C
      DOUBLE PRECISION CPUTOT
C                ! total cpu seconds used in case [s]
C
      DOUBLE PRECISION CQFISH
      DOUBLE PRECISION CQLAPH
      DOUBLE PRECISION CQMEAT
      DOUBLE PRECISION CQRND
      DOUBLE PRECISION CQTDOS
      DOUBLE PRECISION CQVEG
      DOUBLE PRECISION CQWAT
      DOUBLE PRECISION CQXIN
C                ! consequence common blocks []
C
      DOUBLE PRECISION VALUE(MXPAR)
C                ! parameter values []
C
      DOUBLE PRECISION SPAIR1
      DOUBLE PRECISION SPAIR2
      DOUBLE PRECISION SPAIR3
      DOUBLE PRECISION SPDEP1
      DOUBLE PRECISION SPDEP2
      DOUBLE PRECISION SPLAKE
      DOUBLE PRECISION SPRATH
      DOUBLE PRECISION SPTIME
C                ! sampled parameter common blocks []
      DOUBLE PRECISION UNUSED
C                ! sampled parameter common block []
C
      DOUBLE PRECISION CPALL
C                ! calculated parameter common block []
C
      INTEGER I
C                ! implied do loop index []
```

```
      INTEGER IE
C                  ! nuclide descriptions pointer []
      INTEGER IPAR
C                  ! do loop index []
      INTEGER IPMOV
C                  ! do loop index []
      INTEGER J
C                  ! parameter value pointer []
      INTEGER K
C                  ! do loop index []
      INTEGER N
C                  ! do loop index []
      INTEGER NACCPT
C                  ! number of accepted runs []
      INTEGER NHOLD
C                  ! number of runs on hold []
      INTEGER NMOVES
C                  ! for removing constants from output []
      INTEGER NR
C                  ! direct access file record number []
      INTEGER NREJCT
C                  ! number of rejected runs []
      INTEGER NUC
C                  ! do loop index []
      INTEGER RUNNUM
C                  ! current run number []
C
      LOGICAL ACCEPT
C                  ! run acceptance flag []
      LOGICAL ONHOLD
C                  ! signal that an error has been []
C                  ! encountered in the current run []
C
      REAL HOLD
C                  ! maximum real value []
      REAL REAL
C                  ! function to convert to real []
C
C.....define all calculated, consequence, and sampled
C.....common blocks as arrays
C
      COMMON /CPALL/
     1 CPALL(MXSTCM)
C
C
      COMMON /CQFISH/
     1 CQFISH(MXSTCM)
C
      COMMON /CQLAPH/
     1 CQLAPH(MXSTCM)
C
      COMMON /CQMEAT/
     1 CQMEAT(MXSTCM)
C
      COMMON /CQRND/
     1 CQRND(MXSTCM)
C
      COMMON /CQTDOS/
     1 CQTDOS(MXSTCM)
C
```

```
      COMMON /CQVEG/
     1 CQVEG(MXSTCM)
C
      COMMON /CQWAT/
     1 CQWAT(MXSTCM)
C
      COMMON /CQXIN/
     1 CQXIN(MXSTCM)
C
C
      COMMON /SPAIR1/
     1 SPAIR1(MXSTCM)
C
      COMMON /SPAIR2/
     1 SPAIR2(MXSTCM)
C
      COMMON /SPAIR3/
     1 SPAIR3(MXSTCM)
C
      COMMON /SPDEP1/
     1 SPDEP1(MXSTCM)
C
      COMMON /SPDEP2/
     1 SPDEP2(MXSTCM)
C
      COMMON /SPLAKE/
     1 SPLAKE(MXSTCM)
C
      COMMON /SPRATH/
     1 SPRATH(MXSTCM)
C
      COMMON /SPTIME/
     1 SPTIME(MXSTCM)
C
      COMMON /UNUSED/
     1 UNUSED(MXSTCM)
C
      DATA
     1 HOLD/1.7014117E+38/
C
C.....get parameter values in input order (with all consequences
C.....last) from correct common block locations
      J = 0
      DO 20 N = 1,NUMMTH
         DO 10 K = PSTART(N),PEND(N)
            J = J + 1
            IF (COMNAM(COMIDX(K)) .EQ. 'SPAIR1') THEN
               VALUE(J) = SPAIR1(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPAIR2') THEN
               VALUE(J) = SPAIR2(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPAIR3') THEN
               VALUE(J) = SPAIR3(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPDEP1') THEN
               VALUE(J) = SPDEP1(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPDEP2') THEN
               VALUE(J) = SPDEP2(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPLAKE') THEN
               VALUE(J) = SPLAKE(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPRATH') THEN
               VALUE(J) = SPRATH(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'SPTIME') THEN
               VALUE(J) = SPTIME(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'UNUSED') THEN
               VALUE(J) = UNUSED(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CPALL') THEN
               VALUE(J) = CPALL(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQFISH') THEN
               VALUE(J) = CQFISH(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQLAPH') THEN
               VALUE(J) = CQLAPH(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQMEAT') THEN
               VALUE(J) = CQMEAT(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQRND') THEN
               VALUE(J) = CQRND(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQTDOS') THEN
               VALUE(J) = CQTDOS(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQVEG') THEN
               VALUE(J) = CQVEG(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQWAT') THEN
               VALUE(J) = CQWAT(PARIDX(K))
            ELSE IF (COMNAM(COMIDX(K)) .EQ. 'CQXIN') THEN
               VALUE(J) = CQXIN(PARIDX(K))
            END IF
   10    CONTINUE
   20 CONTINUE
      IF (INDPAR) THEN
C........write parameter values optional output
         WRITE (IUPAR,30) RUNNUM,NACCPT
   30    FORMAT('1','RUN NUMBER ',I5,' (ACCEPTED NUMBER ',I5,')')
         WRITE (IUPAR,40)
   40    FORMAT(/,' SELECTED PARAMETERS',/)
         WRITE (IUPAR,50) (I,PARDES(I),PSNAME(I),VALUE(I),PARUNT(I),
     1      I=1,NSPAR)
   50    FORMAT(I5,1X,A32,1X,A14,1X,1PG14.7,1X,A12)
         WRITE (IUPAR,60)
   60    FORMAT(/,' CONSEQUENCES',/)
         IF (ONHOLD) THEN
            WRITE (IUPAR,70)
   70       FORMAT(' NOT DETERMINED - RUN ON HOLD')
         ELSE
            DO 100 I = NSPAR+1,NSPAR+NUMCQ
               IF (PARTYP(I) .EQ. 'LIT') THEN
                  WRITE (IUPAR,80) I,PARDES(I),VALUE(I),PARUNT(I)
   80             FORMAT(1X,I4,2X,A32,4X,A4,2X,A12)
               ELSE
                  WRITE (IUPAR,90) I,PARDES(I),VALUE(I),PARUNT(I)
   90             FORMAT(1X,I4,2X,A32,2X,1PD14.7,2X,A12)
               END IF
  100       CONTINUE
         END IF
      END IF
C*********************************************************
C
C  KLUDGE ENTRY TO SAVE STUFF FOR STATISTICAL CALCULATIONS
C
      CALL BGSAVE (VALUE)
C*********************************************************
      IF (OUTTYP .EQ. 'LONG ') THEN
C........write non-constant values to long output file
         NMOVES = 0
         DO 120 IPAR = 1,NSPAR
            IF (DSTTPS(IPAR) .EQ. 'CONST') THEN
               DO 110 IPMOV = IPAR-NMOVES,NSPAR+NUMCQ-NMOVES-1
                  VALUE(IPMOV) = VALUE(IPMOV+1)
110            CONTINUE
               NMOVES = NMOVES + 1
            END IF
120      CONTINUE
         NR = VARBGN + RUNNUM - 1
         IF (ONHOLD) THEN
            WRITE (IUOUT,REC=NR) RUNNUM,(REAL(VALUE(I)),I=1,NPVAR),
     1      (HOLD,I=1,NUMCQ)
         ELSE
            WRITE (IUOUT,REC=NR) RUNNUM,(REAL(VALUE(I)),I=1,NPVAR+NUMCQ)
         END IF
         IF (ACCEPT) THEN
C...........update nuclide usage
            NR = NUCBGN
            DO 130 NUC = 1,NNUC,NNDFLD
               IE = MIN(NUC+NNDFLD-1,NNUC)
               WRITE (IUOUT,REC=NR) (NUCID(I) // '  ',
     1         NUCUSE(I),I=NUC,IE)
               NR = NR + 1
130         CONTINUE
         END IF
         CPUTOT = CPUCAS + CPUTIM
C.......update run control information
         WRITE (IUOUT,REC=2) RUNNUM,NACCPT,NREJCT,NHOLD,CPUTOT
      ELSE IF (OUTTYP .EQ. 'SHORT') THEN
         IF (ACCEPT) THEN
C...........write saved values to short output file
            NMOVES = 0
            DO 140 IPAR = 1,NSPAR+NUMCQ
               IF (PARSAV(IPAR) .EQ. 'S') THEN
                  NMOVES = NMOVES + 1
                  VALUE(NMOVES) = VALUE(IPAR)
               END IF
140         CONTINUE
            IF (ONHOLD) THEN
               WRITE (IUACC) RUNNUM,(REAL(VALUE(I)),I=1,NSAVPR),
     1         (HOLD,I=1,NSAVCQ)
            ELSE
               WRITE (IUACC) RUNNUM,(REAL(VALUE(I)),I=1,NSAVPR+NSAVCQ)
            END IF
C...........update nuclide usage
            NR = NUCBGN
            DO 150 NUC = 1,NNUC,NNDFLD
               IE = MIN(NUC+NNDFLD-1,NNUC)
               WRITE (IURST,REC=NR) (NUCID(I) // '  ',
     1         NUCUSE(I),I=NUC,IE)
               NR = NR + 1
150         CONTINUE
         ELSE
C...........write run number to .REJ file
            WRITE (IUREJ,*) RUNNUM
         END IF
         CPUTOT = CPUCAS + CPUTIM
C.......update run control information
         WRITE (IURST,REC=2) RUNNUM,NACCPT,NREJCT,NHOLD,CPUTOT
      END IF
      RETURN
      END
      SUBROUTINE ZCALAK(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C**************************************************************** ZCALAK
C
C Calculate CALAKE from CAMIX.
C
C Precondition:  There is at least 1 time series on stack, CAMIX.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values obtained by applying
C                      processes 4.3.4 and 4.3.2 to CAMIX on the
C                      stack.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C***********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION CALNOW
C                ! current value of CALIN [mol/m**3]
      DOUBLE PRECISION CALPRV
C                ! value of CALIN at previous time step [mol/m**3]
      DOUBLE PRECISION EX
C                ! intermediate exponential value [-]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
      DOUBLE PRECISION BGCON
C                ! background concentration of sulphate [mol/m**3]
      DOUBLE PRECISION ZERO
C                ! constant 0.D0
C
      INTEGER J
```

```
C              ! general loop index []
      INTEGER NT
C              ! number of times in subseries []
      INTEGER ONE
C              ! 1 []
C
      LOGICAL CHKSTK
C              ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C              ! continuation flag []
      LOGICAL OK
C              ! ZCALAK call completed ok []
      LOGICAL STOPP
C              ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZCALAK '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES CAMIX NEEDED ON STACK TO COMPUTE CALAKE'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C           WRITE(*,900) MODNAM
C900        FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........Make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 10 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
10          CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of CALAKE for times passed down using
C...........finite difference approach.  First get values for CAMIX.
            CALL INTETS(NT,TIM,ONE,VAL,OK)
            IF (OK) THEN
C..............Initialize CALNOW and BGCON.
               BGCON = 0.1D0
               CALNOW = BGCON
               DO 20 J = 1,NT
C..................invariant: J-1 values of CALAKE computed.  CALNOW
C..................stores the value of CALIN at TIM(J-1).
                  CALPRV = CALNOW
                  CALNOW = VAL(J) * 2.5D-2 + BGCON
                  IF (TIM(J) .LE. ZERO) THEN
                     EX     = EXP( - DELT/XLAML )
                     VAL(J) = BGCON*EX + CALNOW*(1.D0-EX)
                  ELSE
                     EX     = EXP( - (TIM(J)-TIM(J-1)) / XLAML )
                     VAL(J) = VAL(J-1)*EX + CALNOW*(1.D0-EX)
                  END IF
20             CONTINUE
            END IF
C...........Make sure this routine is not called again with more times,
C...........as this algorithm is not repeatable.
            CONTIN = .FALSE.
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZCAMIX(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C******************************************************************* ZCAMIX
C
C Calculate CAMIX from CONCS8, ION, and GYPLFT on stack.
C
C Precondition:  There are at least 3 time series on stack, in order
C                   from the top of stack, GYPLFT, ION, CONCCA.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values for CAMIX obtained by
C                      application of process 1.2.3.6.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
C
      CHARACTER CALTYP*(*)
C               ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C               ! module name []
      CHARACTER MSG*53
C               ! error message []
C
      DOUBLE PRECISION TIM(*)
C               ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C               ! values of sum in part of
C               ! time series [quantity/time]
      DOUBLE PRECISION ZERO
C               ! 0.D0 []
C
      INTEGER J
C               ! general loop index []
      INTEGER NT
C               ! number of times in subseries []
      INTEGER ONE
C               ! 1 []
      INTEGER TWO
```

```
C                 ! 2 []
      INTEGER THREE
C                 ! 3 []
C
      LOGICAL CHKSTK
C                 ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                 ! continuation flag []
      LOGICAL OK
C                 ! ZCAMIX call completed ok []
      LOGICAL STOPP
C                 ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZCAMIX '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 TWO/2/,
     1 THREE/3/,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(THREE)
      IF (.NOT.OK) THEN
         MSG = 'THREE TIME SERIES NEEDED ON STACK TO CALCULATE CAMIX'
         CALL WRERR(MODNAM,MSG.STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C            WRITE(*,900) MODNAM
C900         FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........Determine initial times of CAMIX time series.
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 5 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
5           CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of CAMIX for times passed down.
C...........First interpolate in time series.  VAL = CONCCA;
C...........VALS1 = ION; VALS2 = GYPLFT.
            CALL INTETS(NT,TIM,THREE,VAL,OK)
            IF (OK) CALL INTETS(NT,TIM,TWO,VALS1,OK)
            IF (OK) CALL INTETS(NT,TIM,ONE,VALS2,OK)
C...........Compute CAMIX.
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values computed.
                  IF (TIM(J) .LE. TFINAL ) THEN
                     CONTINUE
                  ELSE IF (VALS2(J) .GT. ZERO) THEN
                     VAL(J) = VALS1(J)
                  ELSE
                     VAL(J) = ZERO
                  END IF
10             CONTINUE
            END IF
         ELSE
C...........unidentified call type
```

```
          MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1    ' SHOULD BE TIMES OR VALUES'
          CALL WRERR(MODNAM,MSG,STOPP)
          OK = .FALSE.
        END IF
      END IF
      RETURN
      END
      SUBROUTINE ZCLAKE(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C************************************************************ ZCLAKE
C
C Calculate CLAKE from MOLTOB*RAMIX according to process 4.1.2 and 4.1.1.
C
C Precondition:  There is at least 1 time series on stack, MOLTOB*RAMIX.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                      intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of CLAKE obtained
C                      by applying processes 4.1.2 and 4.1.1 to
C                      MOLTOB*RAMIX.
C                   CONTIN = .FALSE. so that the call is not repeated.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C*******************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'H:OUTOPT.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'SPLAKE.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION A
C                ! intermediate variable [Bq]
      DOUBLE PRECISION A11
C                ! intermediate variable [Bq/a]
      DOUBLE PRECISION A21
C                ! intermediate constant [Bq/a]
      DOUBLE PRECISION A12
C                ! intermediate constant [1/a]
      DOUBLE PRECISION A22
C                ! intermediate constant [1/a]
      DOUBLE PRECISION A13
```

```
C                 ! intermediate constant [1/a]
      DOUBLE PRECISION A23
C                 ! intermediate constant [1/a]
      DOUBLE PRECISION B
C                 ! intermediate variable [Bq]
      DOUBLE PRECISION B1
C                 ! intermediate constant [1/a]
      DOUBLE PRECISION B2
C                 ! intermediate constant [1/a]
      DOUBLE PRECISION B3
C                 ! intermediate constant [1/a]
      DOUBLE PRECISION DELTAT
C                 ! time step from J-1 to J; usually = DELT [a]
      DOUBLE PRECISION DENOM
C                 ! intermediate constant [1/a**2]
      DOUBLE PRECISION KS
C                 ! intermediate variable [Bq]
      DOUBLE PRECISION KW
C                 ! intermediate variable [Bq]
      DOUBLE PRECISION LAM1
C                 ! intermediate constant [1/a]
      DOUBLE PRECISION LAM2
C                 ! intermediate constant [1/a]
      DOUBLE PRECISION MLAKE
C                 ! previous value of mass of Ra in lake water [Bq]
      DOUBLE PRECISION MSED
C                 ! previous value of mass of Ra in lake sediment [Bq]
      DOUBLE PRECISION TIM(*)
C                 ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                 ! values of sum in part of
C                 ! time series [quantity/time]
      DOUBLE PRECISION XK2
C                 ! rate of Ra gain from sediment [1/a]
      DOUBLE PRECISION XT
C                 ! intermediate variable [Bq]
      DOUBLE PRECISION YT
C                 ! intermediate variable [Bq]
      DOUBLE PRECISION ZERO
C                 ! 0.D0 []
C
      INTEGER J
C                 ! general loop index []
      INTEGER NT
C                 ! number of times in subseries []
      INTEGER ONE
C                 ! 1 []
C
      LOGICAL CHKSTK
C                 ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                 ! continuation flag []
      LOGICAL OK
C                 ! ZCLAKE call completed ok []
      LOGICAL STOPP
C                 ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZCLAKE '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES NEEDED ON STACK TO COMPUTE CLAKE'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C...........Determine initial times of CLAKE time series.
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 10 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
10          CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of CLAKE for times passed down.
C...........First interpolate in time series.  VALS1 = MOLTOB*RAMIX.
            CALL INTETS(NT,TIM,ONE,VALS1,OK)
C...........Compute CLAKE.
            IF (OK) THEN
C..............Evaluate the constants used.
               XK2 = XK1 * BETA
               A21   = ZERO
               A12   = XK1 + 1.D0/XLAML
               A22   = XK1
               A13   = XK2
               A23   = XK2 + XK3
               DENOM = A12*A23 - A22*A13
               B1    = A13 + A23
               B2    = A12 + A22
               B3    = SQRT( (A12+A23)**2 - 4.D0*DENOM )
               LAM1  = ( -A12-A23+B3 ) / 2.D0
               LAM2  = ( -A12-A23-B3 ) / 2.D0
               DO 20 J = 1,NT
C.................invariant: J-1 values computed.
                  IF (TIM(J) .LE. ZERO .OR. J .EQ. 1 ) THEN
                     DELTAT = DELT
                     A11    = ATAIL * NPRCP * VALS1(1)
                     MLAKE  = KW
                     MSED   = KS
                  ELSE
                     DELTAT = TIM(J) - TIM(J-1)
                     A11    = ATAIL * NPRCP * VALS1(J-1)
                  END IF
                  KS     = (A12*A21 + A11*A22) / DENOM
                  KW     = (A13*A21 + A11*A23) / DENOM
                  XT     = MLAKE - KW
                  YT     = MSED - KS
                  A      =   (B1+LAM1) *
     1                    ( (B2+LAM2)*XT - (B1+LAM2)*YT ) /
     1                    ( B3 * (B2-B1) )
                  B      = (B2+LAM1) * A / (B1+LAM1)
                  MLAKE  = KW + A*EXP(LAM1*DELTAT) +
     1                    (XT-A)*EXP(LAM2*DELTAT)
                  MSED   = KS + B*EXP(LAM1*DELTAT) +
     1                    (YT-B)*EXP(LAM2*DELTAT)
```

```
                  IF (INDSUB) WRITE(IUSUB,905) J,TIM(J),MSED
905               FORMAT(' ** ZCLAKE ** J = ',I6,', T = ',1PG13.4,
     1               ', MSED = ',G13.4)
C.................transform lake mass to concentration
                  VAL(J) = MLAKE / VLAKE
20             CONTINUE
            END IF
C...........stop further calls with other times
            CONTIN = .FALSE.
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZCONCC(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C***************************************************************** ZCONCC
C
C Calculate CONCCA from CONCS8.
C
C Precondition:  There is at least 1 time series, CONCS8, on stack.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values obtained by applying
C                      process 1.2.3.2 to CONCS8 on the stack.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C***********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
      INTEGER ONE
C                ! 1 []
C
      LOGICAL CHKSTK
C                ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL OK
```

```
C                ! ZCONCC call completed ok []
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZCONCC '/,
     1 ONE/1/,
     1 STOPP/.TRUE./
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES CONCSB NEEDED ON STACK TO COMPUTE CONCCA'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C            WRITE(*,900) MODNAM
C900         FORMAT(' ** ',A6,' ** CALLED WITH CALTYP = TIMES')
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 5 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
5           CONTINUE
C            WRITE(*,903) (J,TIM(J),J = 1,NT)
C903         FORMAT(' ** ZCONCC ** RETURNING WITH TIMES ',(I6,G12.4))
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C            WRITE(*,901) NT,(J,TIM(J),J=1,NT)
C901         FORMAT(' ** ZCONCC ** CALLED WITH CALTYP = VALUES, NT =',I7,
C    1           /,' AND TIMES ',(I6,G12.4))
C...........determine values of CONCCA for times passed down
            CALL INTETS(NT,TIM,ONE,VAL,OK)
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values of CONCCA computed
                  VAL(J) = SQRT( AEQBM**2 + (VAL(J)/2.D0)**2 ) -
     1               VAL(J)/2.D0
10             CONTINUE
            END IF
C            WRITE(*,902) (J,VAL(J),J = 1,NT)
C902         FORMAT(' ** ZCONCC ** RETURNING WITH VALUES ',(I6,G12.4))
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE.' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZCONCS(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C****************************************************************** ZCONCS
C
C Calculate CONCS8.
C
C Precondition:  IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of CONCS8 at TIM(1:NT).
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
C
      LOGICAL OK
C                ! ZCONCS call completed ok []
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZCONCS '/,
     1 STOPP/.TRUE./
C
      CONTIN = .TRUE.
      OK     = .TRUE.
      IF (CALTYP .EQ. 'TIMES') THEN
C         WRITE(*,900) MODNAM
C900      FORMAT(' ** ',A6,' ** CALLED WITH CALTYP = TIMES')
C...........make up a uniform mesh of times based on DELT
         NT = 1 + INT(TLIMIT/DELT)
         DO 5 J = 1,NT
            TIM(J) = DELT * DBLE(J-1)
5        CONTINUE
      ELSE IF (CALTYP .EQ. 'VALUES') THEN
C........compute values of CONCS8
         DO 10 J = 1,NT
C........invariant: J-1 values have been computed
            VAL(J) = CSINIT * EXP(-KRATE*TIM(J))
10       CONTINUE
      ELSE
C........unidentified call type
         MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1         ' SHOULD BE TIMES OR VALUES'
         CALL WRERR(MODNAM,MSG,STOPP)
         OK = .FALSE.
      END IF
      RETURN
      END
      SUBROUTINE ZCRAAV(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C******************************************************************* ZCRAAV
C
C Calculate CRAAVG from CONCCA, FRACRA, and RAFTH on stack, according to
C    process 1.5.1.
C
C Precondition:  There are at least 3 time series on stack, in order
C                   from the top of stack, RAFTH, FRACRA, CONCCA.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values for CRAAVG obtained by
C                      application of process 1.5.1.
C                END IF
C
C  86-APR-10  VERSION 01   T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
      DOUBLE PRECISION ZERO
C                ! 0.D0 []
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
      INTEGER ONE
```

```
C                ! 1 []
      INTEGER TWO
C                ! 2 []
      INTEGER THREE
C                ! 3 []
C
      LOGICAL CHKSTK
C                ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL OK
C                ! ZCRAAV call completed ok []
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZCRAAV '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 TWO/2/,
     1 THREE/3/,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(THREE)
      IF (.NOT.OK) THEN
         MSG = 'THREE TIME SERIES NEEDED ON STACK TO CALCULATE CRAAVG'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C...........Determine initial times of CRAAVG time series.
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 5 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
5           CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of CRAAVG for times passed down.
C...........First interpolate in time series.  VALS1 = CONCCA;
C...........VALS2 = FRACRA; VALS3 = RAFTH.
            CALL INTETS(NT,TIM,THREE,VALS1,OK)
            IF (OK) CALL INTETS(NT,TIM,TWO,VALS2,OK)
            IF (OK) CALL INTETS(NT,TIM,ONE,VALS3,OK)
C...........Compute CRAAVG.
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values computed.
                  IF (TIM(J) .LT. TFINAL ) THEN
                     VAL(J) = VALS3(J) + DISTCO*VALS2(J)*VALS1(J)
                  ELSE
                     VAL(J) = VALS3(J)
                  END IF
10             CONTINUE
            END IF
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
```

```
         CALL WRERR(MODNAM,MSG,STOPP)
         OK = .FALSE.
      END IF
   END IF
   RETURN
   END
```

```
      SUBROUTINE ZDEPT1(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C**************************************************************** ZDEPT1
C
C Calculate DEPTH1 according to process 1.2.4.1.
C
C Precondition:  IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of DEPTH1 at TIM(1:NT).
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C***********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'D1COM.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION ACCUR
C                ! precision in root finding with BISECT (a small
C                ! fraction [-]
      DOUBLE PRECISION HILIM
C                ! high limit [m]
      DOUBLE PRECISION LOWLIM
C                ! low limit [m]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
C
      EXTERNAL DEPFNC
C
      INTEGER IERR
C                ! error condition indicator; 0 is good [-]
      INTEGER J
C                ! general loop index []
```

```
      INTEGER NT
C                   ! number of times in subseries []
C
      LOGICAL OK
C                   ! ZDEPT1 call completed ok []
      LOGICAL CONTIN
C                   ! continuation flag []
      LOGICAL STOPP
C                   ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZDEPT1 '/,
     1 STOPP/.TRUE./
C
      CONTIN = .TRUE.
      OK     = .TRUE.
      IF (CALTYP .EQ. 'TIMES') THEN
C.......make up a uniform mesh of times based on DELT
         NT = 1 + INT(TLIMIT/DELT)
         DO 5 J = 1,NT
            TIM(J) = DELT * DBLE(J-1)
5        CONTINUE
      ELSE IF (CALTYP .EQ. 'VALUES') THEN
C.......compute values of DEPTH1
         DO 20 J = 1,NT
C..........invariant: J-1 values have been computed
            IF ( TIM(J) .LT. TFINAL ) THEN
C.............BISECT will locate a root of DEPFNC.  The first three
C.............arguments of BISECT are the lower and upper limits of
C.............a range containing the root, and the accuracy desired
C.............in the root.  Time T is passed in D1COM.
               LOWLIM = 0.D0
               HILIM  = DUNSAT
               ACCUR  = 1.0D-5
               T      = TIM(J)
               CALL BISECT(LOWLIM,HILIM,ACCUR,DEPFNC,VAL(J),IERR)
               IF (IERR.GT.0) THEN
                  WRITE(MSG,10) IERR
10                FORMAT('ERROR IN FINDING ROOT IN BISECT; IERR = ',
     1               I4)
                  CALL WRERR(MODNAM,MSG,STOPP)
                  OK = .FALSE.
               END IF
            ELSE
               VAL(J) = DUNSAT
            END IF
20       CONTINUE
      ELSE
C.......unidentified call type
         MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1         ' SHOULD BE TIMES OR VALUES'
         CALL WRERR(MODNAM,MSG,STOPP)
         OK = .FALSE.
      END IF
      RETURN
      END
      SUBROUTINE ZEMRAA(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C*************************************************************** ZEMRAA
C
C Calculate EMRARA from MOLTOB*RACON1 according to process 2.1.
C   Note: EMRARA is also known as EMRATE(RA).
C
C Precondition:  There is at least 1 time series on stack, MOLTOB*RACON1
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains all the times from the top
C                      time series on the stack (MOLTOB*RACON1).
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of EMRATE(Ra) obtained
C                      by applying process 2.1 to MOLTOB*RACON1 on
C                      the stack.
C                END IF
C
C  86-MAR-19  VERSION 01  T. ANDRES
C
C*********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'SPDEP.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION CONST
C                ! constant used in calculating EMRARA [m**3/s]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
      DOUBLE PRECISION ZERO
C                ! 0.D0 [-]
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
      INTEGER ONE
C                ! 1 []
```

```
C
      LOGICAL CHKSTK
C                ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL OK
C                ! ZEMRAA call completed ok []
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZEMRAA '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES MOLTOB*RACON1 NEEDED TO COMPUTE EMRARA'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C...........copy times of time series MOLTOB*RACON1
            NT = 0
            CALL RETIMS(ONE,ZERO,TLIMIT,NT,TIM,OK)
C...........initialize a constant used for every value.
            CONST  = 2.2D-6 * SILT * (365.D0 -FPREC) * FWIND1 * AEXPT /
     1               (1.5 * 235.D0 * 15.D0 * RHOTAL)
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........determine values of EMRARA for times passed down.  First
C...........retrieve values of MOLTOB*RACON1.
            CALL INTETS(NT,TIM,ONE,VAL,OK)
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values of EMRARA computed
                  VAL(J) = CONST * VAL(J)
10             CONTINUE
            END IF
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZEMRAN(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C*********************************************************************** ZEMRAN
C
C Calculate EMRARN from DEPTH1, REPRA, and MOLTOB*RACON1 on stack,
C    according to processes 1.2.4.3, 1.6, 2.2.2, and 2.2.1.
C    Note:  EMRARN is identical to EMRATE(RN).
C
C Precondition:  There are at least 3 time series on stack, in order
C                   from the top of stack, MOLTOB*RACON1, REPRA, DEPTH1.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values for EMRARN obtained by
C                      application of processes 1.2.4.3, 1.6, 2.2.2,
C                      and 2.2.1.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C***********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
C
      CHARACTER CALTYP*(*)
C               ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C               ! module name []
      CHARACTER MSG*53
C               ! error message []
C
      DOUBLE PRECISION COTH2
C               ! current value of hyperbolic cotangent [-]
      DOUBLE PRECISION DEPTH1
C               ! current value of depth of upper unsat. zone [m]
      DOUBLE PRECISION DEPTH2
C               ! current value of depth of lower unsat. zone [m]
      DOUBLE PRECISION MRACN1
C               ! current value of MOLTOB*RACON1, Ra conc. in upper
C               ! unsat. zone [Bq/m**3]
      DOUBLE PRECISION MRACN2
C               ! current value of MOLTOB*RACON2, Ra conc. in lower
```

```
C                   ! unsat. zone [Bq/m**3]
      DOUBLE PRECISION RNFLUX
C                   ! current value of Rn flux from the entire unsaturated
C                   ! zone [Bq/m**2/s]
      DOUBLE PRECISION S1
C                   ! current value of Rn activity emanating from the
C                   ! upper unsat. zone [Bq/m**3/s]
      DOUBLE PRECISION S2
C                   ! current value of Rn activity emanating from the
C                   ! lower unsat. zone [Bq/m**3/s]
      DOUBLE PRECISION TERM
C                   ! intermediate value; term that includes an
C                   ! indeterminate form coth(x)/coth(x) [Bq/m**3/s]
      DOUBLE PRECISION TIM(*)
C                   ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                   ! values of sum in part of
C                   ! time series [quantity/time]
      DOUBLE PRECISION ZERO
C                   ! 0.D0 []
C
      INTEGER J
C                   ! general loop index []
      INTEGER NT
C                   ! number of times in subseries []
      INTEGER ONE
C                   ! 1 []
      INTEGER TWO
C                   ! 2 []
      INTEGER THREE
C                   ! 3 []
C
      LOGICAL CHKSTK
C                   ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                   ! continuation flag []
      LOGICAL OK
C                   ! ZEMRAN call completed ok []
      LOGICAL STOPP
C                   ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZEMRAN '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 TWO/2/,
     1 THREE/3/,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(THREE)
      IF (.NOT.OK) THEN
         MSG = 'THREE TIME SERIES NEEDED ON STACK TO CALCULATE EMRARN'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C...........Determine initial times of EMRARN time series.
C...........make up a uniform mesh of times based on DELT
```

```
            NT = 1 + INT(TLIMIT/DELT)
            DO 5 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
5           CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of EMRARN for times passed down.
C...........First interpolate in time series.  VALS1 = DEPTH1;
C...........VALS2 = REPRA; VALS3 = MOLTOB*RACON1.
            CALL INTETS(NT,TIM,THREE,VALS1,OK)
            IF (OK) CALL INTETS(NT,TIM,TWO,VALS2,OK)
            IF (OK) CALL INTETS(NT,TIM,ONE,VALS3,OK)
C...........Compute EMRARN.
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values computed.
C.................First calculate DEPTH2 from process 1.2.4.3 and
C.................MOLTOB*RACON2 from process 1.6, modified to take
C.................MOLTOB*RACON1 as input, rather than RACON1.
                  MRACN1 = VALS3(J)
                  DEPTH1 = VALS1(J)
                  IF (TIM(J) .LT. TFINAL ) THEN
                     DEPTH2 = DUNSAT - DEPTH1
                     MRACN2 = MRACN1 +
     1                        MOLTOB * (CRAGIN + VALS2(J)/DEPTH2)
                  ELSE
                     DEPTH2 = ZERO
                     MRACN2 = MRACN1
                  END IF
C.................Calculate S1 and S2 from process 2.2.2.
                  S1 = MRACN1 * RNLAM * EMANAT
                  S2 = MRACN2 * RNLAM * EMANAT
C.................Calculate RNFLUX and EMRARN [= EMRATE(Rn)] according
C.................to process 2.2.1.
                  IF ( DEPTH2 .GT. 0.D0 ) THEN
                     COTH2  = 1.D0 / TANH(DIFLEN*DEPTH2)
                     TERM   =  (  S2 - S1 +
     1                             (1.D0 + COTH2)*S1*EXP(DIFLEN*DEPTH1)
     1                        ) / ( TANH(DIFLEN*DEPTH1) + COTH2 )
                  ELSE
                     TERM   = S1 * EXP(DIFLEN*DEPTH1)
                  END IF
                  RNFLUX = -S1 / DIFLEN +
     1                       TERM / ( DIFLEN * COSH(DIFLEN*DEPTH1) )
                  VAL(J) = RNFLUX * ATAIL
10             CONTINUE
            END IF
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZFRACR(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C******************************************************************* ZFRACR
C
C Calculate FRACRA from CONCCA, DEPTH1, REPGYP, and RAFTH on stack
C    according to processes 1.2.4.2 and 1.4.1.
C
C Precondition:  There are at least 4 time series on stack, in order
C                   from the top of stack, CONCCA, DEPTH1, REPGYP, and
C                   RAFTH.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                      intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values obtained by applying
C                      process 1.4.1 to time series on the stack.
C                   CONTIN = .FALSE. so that the call will not be
C                      repeated.
C                END IF
C
C  86-MAY-07  VERSION 01  T. ANDRES
C
C**************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'FRCCOM.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
C
      CHARACTER CALTYP*(*)
C               ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C               ! module name []
      CHARACTER MSG*53
C               ! error message []
C
      DOUBLE PRECISION ACCUR
C               ! precision indicator (small fraction) [-]
C      DOUBLE PRECISION FRCFNC
C               ! name of function of which FRACRA is a zero [-]
      DOUBLE PRECISION HILIM
C               ! high limit on FRACRA [-]
      DOUBLE PRECISION LOWLIM
C               ! low limit on FRACRA [-]
      DOUBLE PRECISION TIM(*)
C               ! times in part of time series [time]
```

```
      DOUBLE PRECISION VAL(*)
C               ! values of sum in part of
C               ! time series [quantity/time]
      DOUBLE PRECISION ZERO
C               ! 0.D0 []
C
      EXTERNAL FRCFNC
C               ! evaluate function of which FRACRA is a zero [-]
C
      INTEGER IERR
C               ! error condition indicator: 0 is good [-]
      INTEGER J
C               ! general loop index []
      INTEGER NT
C               ! number of times in subseries []
      INTEGER ONE
C               ! 1 []
      INTEGER TWO
C               ! 2 []
      INTEGER THREE
C               ! 3 []
      INTEGER FOUR
C               ! 4 []
C
      LOGICAL OK
C               ! ZFRACR call completed ok []
      LOGICAL CHKSTK
C               ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C               ! continuation flag []
      LOGICAL STOPP
C               ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZFRACR '/,
     1 ONE/1/,
     1 STOPP/.FALSE./,
     1 TWO/2/,
     1 THREE/3/,
     1 FOUR/4/,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(FOUR)
      IF (.NOT.OK) THEN
         MSG = 'FOUR TIME SERIES NEEDED ON STACK TO CALCULATE FRACRA'
         CALL WRERR(MODNAM,MSG,STOPP)
         OK = .FALSE.
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C           WRITE(*,900) MODNAM
C900        FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........Determine initial times of FRACRA time series.
C...........make up a uniform mesh of times based on DELT
           NT = 1 + INT(TLIMIT/DELT)
           DO 10 J = 1,NT
             TIM(J) = DELT * DBLE(J-1)
10         CONTINUE
```

```
C            WRITE(*,901) MODNAM,NT,(TIM(J),J=1,NT)
C901         FORMAT(' ** ',A6,' ** RETURNING NT = ',I6,', AND TIM = ',/,
C     1         (14X,1P3G12.4))
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C            WRITE(*,902) MODNAM,NT,(TIM(J),J=1,NT)
C902         FORMAT(' ** ',A6,' ** CALLED WITH CALTYP = VALUES',/,
C     1                           NT =',I7,', TIMES = ',/,
C     1         (14X,1P3G12.4))
C...........Determine values of FRACRA for times passed down.
C...........First interpolate in time series.  VALS1 = CONCCA;
C...........VALS2 = DEPTH1; VALS3 = REPGYP; VALS4 = RAFTH.
            CALL INTETS(NT,TIM,FOUR,VALS1,OK)
            IF (OK) CALL INTETS(NT,TIM,THREE,VALS2,OK)
            IF (OK) CALL INTETS(NT,TIM,TWO,  VALS3,OK)
            IF (OK) CALL INTETS(NT,TIM,ONE,  VALS4,OK)
C...........Compute FRACRA.
            IF (OK) THEN
               DO 30 J = 1,NT
C.................invariant: J-1 values computed.
C.................BISECT will locate a root of FRCFNC.  The first three
C.................arguments of BISECT are the lower and upper limits of
C.................a range containing the root, and the accuracy desired
C.................in the root.  Accuracy required is 1 in 10**6.
                  IF ( TIM(J) .LE. ZERO .OR. J .EQ. 1 ) THEN
                     VAL(J) = RAINGP * RADIUM / GYPBLK
                  ELSE IF ( TIM(J) .GE. TFINAL ) THEN
                     VAL(J) = ZERO
                  ELSE
                     CURIND = J
                     DELTAT = TIM(J) - TIM(J-1)
                     PRVFR  = VAL(J-1)
C....................note that bisection is applied to log FRACRA
                     LOWLIM = -20.D0
                     HILIM  = 0.D0
                     ACCUR  = 1.0D-6
                     CALL BISECT (LOWLIM,HILIM,ACCUR,FRCFNC,VAL(J),IERR)
                     VAL(J) = 10.D0**VAL(J)
                     IF (IERR.GT.0) THEN
                        WRITE(MSG,20) IERR
20                      FORMAT('ERROR IN FINDING ROOT IN BISECT; IERR ='
     1                     ,I5)
                        CALL WRERR(MODNAM,MSG,STOPP)
                     END IF
                  END IF
30             CONTINUE
C..............make sure that this part of the code is not called again,
C..............as it is not repeatable.
               CONTIN = .FALSE.
            END IF
C            WRITE(*,903) MODNAM,NT,OK,(VAL(J),J=1,NT)
C903         FORMAT(' ** ',A6,' ** RETURNING NT = ',I6,', OK = ',L6,/,
C     1                           VAL = ',(T20,1P3G12.4))
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZGYPLF(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C******************************************************************* ZGYPLF
C
C Calculate GYPLFT from ION.
C
C Precondition:  There is at least 1 time series, ION, on stack.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                      intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values obtained by applying
C                      process 1.2.3.4 to ION on the stack.
C                   CONTIN = .FALSE. so that the call will not be
C                      repeated.
C                END IF
C
C  86-APR-02  VERSION 01  T. ANDRES
C
C************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'SPDEP.INC'
      INCLUDE 'CPALL.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C               ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C               ! module name []
      CHARACTER MSG*53
C               ! error message []
C
      DOUBLE PRECISION IONNOW
C               ! value of ION at current time [mol/m**3]
      DOUBLE PRECISION IONPRV
C               ! value of ION at previous time [mol/m**3]
      DOUBLE PRECISION MGYPI
C               ! init. mass of gypsum in mixing zone [mol/m**2]
      DOUBLE PRECISION TIM(*)
C               ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C               ! values of sum in part of
C               ! time series [quantity/time]
C
      INTEGER J
C               ! general loop index []
      INTEGER NT
C               ! number of times in subseries []
```

```
      INTEGER ONE
C                    ! 1 []
C
      LOGICAL CHKSTK
C                    ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                    ! continuation flag []
      LOGICAL OK
C                    ! ZGYPLF call completed ok []
      LOGICAL STOPP
C                    ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZGYPLF '/,
     1 ONE/1/,
     1 STOPP/.TRUE./
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES ION NEEDED ON STACK TO COMPUTE GYPLFT'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C           WRITE(*,900) MODNAM
C900        FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 10 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
10          CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of GYPLFT for times passed down using
C...........finite difference approach.  First get values for ION.
            CALL INTETS(NT,TIM,ONE,VAL,OK)
            IF (OK) THEN
C..............Evaluate MGYPI and initialize IONNOW
               MGYPI  = GYPBLK * DMIX
               IONNOW = 0.D0
               DO 20 J = 1,NT
C.................invariant: J-1 values of GYPLFT computed.  IONNOW
C.................stores the value of ION at TIM(J-1).
                  IONPRV = IONNOW
                  IONNOW = VAL(J)
                  IF (TIM(J) .LE. TFINAL) THEN
                     VAL(J) = MGYPI
                  ELSE
                     VAL(J) = VAL(J-1) - (TIM(J)-TIM(J-1)) *
     1                  NPRCP * (IONNOW+IONPRV) / 2.D0
                     VAL(J) = MAX(VAL(J),0.D0)
                  END IF
20             CONTINUE
            END IF
C...........Make sure this routine is not called again with more times,
C...........as this algorithm is not repeatable.
            CONTIN = .FALSE.
         ELSE
C...........unidentified call type
```

```
         MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1         ' SHOULD BE TIMES OR VALUES'
         CALL WRERR(MODNAM,MSG,STOPP)
         OK = .FALSE.
      END IF
   END IF
   RETURN
   END
```

```
      SUBROUTINE ZION(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C******************************************************************* ZION
C
C Calculate ION from CONCS8.
C
C Precondition:  There is at least 1 time series on stack.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values obtained by applying
C                      process 1.2.3.3 to CONCS8 on the stack.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
      INTEGER ONE
C                ! 1 []
C
      LOGICAL CHKSTK
C                ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL OK
```

```
C                ! ZION call completed ok []
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZION '/,
     1 ONE/1/,
     1 STOPP/.TRUE./
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES CONCS8 NEEDED ON STACK TO COMPUTE ION'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C            WRITE(*,900) MODNAM
C900         FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 5 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
5           CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........determine values of ION for times passed down
            CALL INTETS(NT,TIM,ONE,VAL,OK)
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values of ION computed
                  IF (TIM(J) .LE. TFINAL) THEN
                     VAL(J) = 0.D0
                  ELSE
                     VAL(J) = 0.5D0 * (-VAL(J)+
     1                  SQRT( VAL(J)**2 + 4.D0*KSUBSO/FACTCO**2 ) )
                  END IF
10             CONTINUE
            END IF
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZPHLAK(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C********************************************************************* ZPHLAK
C
C Calculate PHLAKE from SULAKE and CALAKE.
C
C Precondition:  There are at least 2 time series on stack, SULAKE and
C                   CALAKE.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of PHLAKE obtained by
C                      applying process 4.3.1 to the top two series
C                      on the time series stack.
C                END IF
C
C  86-MAR-12  VERSION 01  B. GOODWIN
C  86-APR-10  VERSION 02  T. ANDRES
C
C**********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'PHCOM.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
      INCLUDE 'H:TSTACK.INC'
C
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION ACCUR
C                ! precision indicator (small fraction) [-]
      DOUBLE PRECISION HILIM
C                ! high limit on pH [-]
      DOUBLE PRECISION LOWLIM
C                ! low limit on pH [-]
      DOUBLE PRECISION PHFUNC
C                ! name of function of which pH is a zero [-]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
```

```
C                ! values of sum in part of
C                ! time series [quantity/time]
C
      EXTERNAL  PHFUNC
C
C
      INTEGER IERR
C                ! error indicator from BISECT; anything but 0 is bad.
C                ! [-]
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
      INTEGER ONE
C                ! 1 []
      INTEGER TWO
C                ! 2 []
C
      LOGICAL CHKSTK
C                ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL OK
C                ! ZPHLAK call completed ok []
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZPHLAK '/,
     1 ONE/1/,
     1 STOPP/.FALSE./,
     1 TWO/2/
C
C.....check precondition
      OK = CHKSTK(TWO)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES SULAKE AND CALAKE NEEDED ON STACK'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C            WRITE(*,900) MODNAM
C900         FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........Determine initial times of PHLAKE time series.
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 5 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
5           CONTINUE
C            WRITE(*,901) MODNAM,NT,(TIM(J),J=1,NT)
C901         FORMAT(' ** ',A6,' ** RETURNING NT = ',I6,', AND TIM = ',/,
C     1         (14X,1P3G12.4))
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C            WRITE(*,902) MODNAM,NT,(TIM(J),J=1,NT)
C902         FORMAT(' ** ',A6,' ** CALLED WITH CALTYP = VALUES',/,
C     1       '                      NT =',I7,', TIMES = ',/,
C     1         (14X,1P3G12.4))
C...........Determine values of PHLAKE for times passed down.
C...........Interpolate in both time series.  VALS1 = CALAKE;
```

```
C...........VALS2 = SULAKE.
            CALL INTETS(NT,TIM,ONE,VALS1,OK)
            IF (OK) THEN
               CALL INTETS(NT,TIM,TWO,VALS2,OK)
               IF (.NOT. OK) THEN
                  MSG = 'ERROR IN INTERPOLATING SULAKE'
                  CALL WRERR(MODNAM,MSG,STOPP)
               ENDIF
            ELSE
               MSG = 'ERROR IN INTERPOLATING CALAKE'
               CALL WRERR(MODNAM,MSG,STOPP)
            ENDIF
            IF (OK) THEN
               DO 20 J = 1,NT
C.................Invariant: J-1 values of PHLAKE computed.
C.................BISECT will locate a root of PHFUNC.  The first three
C.................arguments of BISECT are the lower and upper limits of
C.................a range containing the root, and the accuracy desired
C.................in the root.  The limits provided here could be
C.................refined if necessary, and correspond to a pH range of
C.................0.0 to 14.0.  Accuracy is 1 in 100000.
                  CALAKE = VALS1(J)
                  SULAKE = VALS2(J)
                  LOWLIM = 0.0D0
                  HILIM  = 14.0D0
                  ACCUR  = 1.0D-5
                  CALL BISECT (LOWLIM,HILIM,ACCUR,PHFUNC,VAL(J),IERR)
                  IF (IERR.GT.0) THEN
                     WRITE(MSG,10) IERR
10                   FORMAT('ERROR IN FINDING ROOT IN BISECT; IERR = ',
     1                  I4)
                     CALL WRERR(MODNAM,MSG,STOPP)
C                     OK = .FALSE.
                  END IF
20             CONTINUE
            END IF
C            WRITE(*,903) MODNAM,NT,OK,(VAL(J),J=1,NT)
C903         FORMAT(' ** ',A6,' ** RETURNING NT = ',I6,', OK = ',L6,/,
C     1                           VAL = ',(T20,1P3G12.4))
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZRACO1(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C************************************************************** ZRACO1
C
C Calculate MOLTOB*RACON1 according to processes 1.3.5 and 1.3.6, with
C    a final multiplication by MOLTOB to convert mols to Bqs.
C
C Precondition:  IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of MOLTOB*RACON1 at
C                      TIM(1:NT).
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C*********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C               ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C               ! module name []
      CHARACTER MSG*53
C               ! error message []
C
      DOUBLE PRECISION C230
C               ! constant used in calculating RATH23 [mol/m**3]
C               ! assumed saved from one call to the next
      DOUBLE PRECISION RATH23
C               ! current value of conc. of Ra-226 due to ingrowth
C               ! from Th-230 [mol/m**3]
      DOUBLE PRECISION TIM(*)
C               ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C               ! values of sum in part of
C               ! time series [quantity/time]
C
      INTEGER J
C               ! general loop index []
      INTEGER NT
C               ! number of times in subseries []
C
```

```
      LOGICAL CONTIN
C                 ! continuation flag []
      LOGICAL OK
C                 ! ZRACO1 call completed ok []
      LOGICAL STOPP
C                 ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZRACO1 '/,
     1 STOPP/.TRUE./
C
      CONTIN = .TRUE.
      OK     = .TRUE.
      IF (CALTYP .EQ. 'TIMES') THEN
C........make up a uniform mesh of times based on DELT
         NT = 1 + INT(TLIMIT/DELT)
         DO 5 J = 1,NT
            TIM(J) = DELT * DBLE(J-1)
5        CONTINUE
C........Evaluate a constant that is used for each value.
         C230   = RAINTH * THFRAC * CTH230 * THLAM /(RALAM - THLAM)
      ELSE IF (CALTYP .EQ. 'VALUES') THEN
C........compute values of MOLTOB*RACON1
         DO 10 J = 1,NT
C........invariant: J-1 values have been computed
            RATH23 = C230 * ( EXP(-THLAM*TIM(J)) - EXP(-RALAM*TIM(J)) )
            VAL(J) = RATH23 + RASIMM * EXP(-RALAM*TIM(J))
            VAL(J) = MOLTOB * VAL(J)
10       CONTINUE
      ELSE
C........unidentified call type
         MSG = 'UNIDENTIFIABLE CALL TYPE.' //
     1       ' SHOULD BE TIMES OR VALUES'
         CALL WRERR(MODNAM,MSG,STOPP)
         OK = .FALSE.
      END IF
      RETURN
      END
      SUBROUTINE ZRAFTH(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C****************************************************************** ZRAFTH
C
C Calculate RAFTH according to process 1.3.7.4 and 1.3.7.5.
C
C Precondition:  IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of RAFTH at TIM(1:NT).
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION THSOLD
C                ! total Th in solid phase [mol/m**3]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
C
      LOGICAL OK
C                ! ZRAFTH call completed ok []
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL STOPP
C                ! signal to stop processing []
```

```
C
      DATA
     1 MODNAM/'ZRAFTH '/,
     1 STOPP/.TRUE./
C
      CONTIN = .TRUE.
      OK     = .TRUE.
      IF (CALTYP .EQ. 'TIMES') THEN
C........make up a uniform mesh of times based on DELT
         NT = 1 + INT(TLIMIT/DELT)
         DO 5 J = 1,NT
            TIM(J) = DELT * DBLE(J-1)
5        CONTINUE
      ELSE IF (CALTYP .EQ. 'VALUES') THEN
C........compute values of RAFTH
         DO 10 J = 1,NT
C...........invariant: J-1 values have been computed
C...........Compute THSOLD as per process 1.3.7.4.
            THSOLD = THINIT *
     1               (  THFRAC +
     1                  (1.DO-THFRAC) * EXP(-THDISS*TIM(J))
     1               )
C...........Compute RAFTH as per process 1.3.7.5.
            VAL(J) = (1.DO-RAINTH)*THLAM*THINTH*THSOLD*DUNSAT/NPRCP
10       CONTINUE
      ELSE
C........unidentified call type
         MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
         CALL WRERR(MODNAM,MSG,STOPP)
         OK = .FALSE.
      END IF
      RETURN
      END
      SUBROUTINE ZRAMIX(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C******************************************************************* ZRAMIX
C
C Calculate MOLTOB*RAMIX from CRAAVG according to process 1.5.2.
C
C Precondition:  There is at least 1 time series on stack, CRAAVG.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                      intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values of MOLTOB*RAMIX obtained
C                      by applying process 1.5.2 to CRAAVG.
C                   CONTIN = .FALSE. so that the call is not repeated.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C*************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'SPLAKE.INC'
      INCLUDE 'SPRATH.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION CURVAL
C                ! current value of RAMIX [mol/m**3]
      DOUBLE PRECISION ECONST
C                ! intermediate constant [-]
      DOUBLE PRECISION PRVAL
C                ! previous value of RAMIX [mol/m**3]
      DOUBLE PRECISION RARET
C                ! Ra retardation const. in mixing zone [-]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
      DOUBLE PRECISION ZERO
```

```
C                 ! 0.DO []
C
        INTEGER J
C                 ! general loop index []
        INTEGER NT
C                 ! number of times in subseries []
        INTEGER ONE
C                 ! 1 []
C
        LOGICAL CHKSTK
C                 ! check that minimum number of ts are in stack [-]
        LOGICAL CONTIN
C                 ! continuation flag []
        LOGICAL OK
C                 ! ZRAMIX call completed ok []
        LOGICAL STOPP
C                 ! signal to stop processing []
C
        DATA
      1 MODNAM/'ZRAMIX '/,
      1 ONE/1/,
      1 STOPP/.TRUE./,
      1 ZERO/0.DO/
C
C.....check precondition
        OK = CHKSTK(ONE)
        IF (.NOT.OK) THEN
           MSG = 'TIME SERIES NEEDED ON STACK TO COMPUTE MOLTOB*RAMIX'
           CALL WRERR(MODNAM,MSG,STOPP)
        END IF
        CONTIN = .TRUE.
        IF (OK) THEN
           IF (CALTYP .EQ. 'TIMES') THEN
C...........Determine initial times of MOLTOB*RAMIX time series.
C...........make up a uniform mesh of times based on DELT
              NT = 1 + INT(TLIMIT/DELT)
              DO 10 J = 1,NT
                 TIM(J) = DELT * DBLE(J-1)
10            CONTINUE
           ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of MOLTOB*RAMIX for times passed down.
C...........First interpolate in time series.  VALS1 = CRAAVG.
              CALL INTETS(NT,TIM,ONE,VALS1,OK)
C...........Compute MOLTOB*RAMIX.
              IF (OK) THEN
C.............Evaluate a constant that is used for each value.
                 RARET = 1.DO + KSUBDR*RHOTAL/TVOID
                 ECONST = EXP( -NPRCP * DELT / (TVOID*DMIX*RARET) )
C.............initial conditions
                 CURVAL = VALS1(1)
                 VAL(1) = CURVAL * MOLTOB
                 PRVAL  = CURVAL
                 DO 20 J = 2,NT
C.................invariant: J-1 values computed.  PRVAL
C.................contains the value of the previous MOLTOB*RAMIX
C.................calculation.
                    CURVAL = VALS1(J-1) +
     1                       ( PRVAL - VALS1(J-1) ) * ECONST
                    VAL(J) = CURVAL * MOLTOB
                    PRVAL  = CURVAL
```

```
20             CONTINUE
            END IF
C...........make sure the call is not repeated with new times.
            CONTIN = .FALSE.
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZREPGY(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C****************************************************************** ZREPGY
C
C Calculate REPGYP from DEPTH1.
C
C Precondition:  There is at least 1 time series on stack.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values obtained by applying
C                      process 1.2.5 to DEPTH1 on the stack.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C************************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'TF.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION TFFUNC
C                ! function to evaluate values of REPGYP [mol/m**2]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
      INTEGER ONE
C                ! 1 []
C
      LOGICAL CHKSTK
C                ! check that minimum number of ts are in stack [-]
```

```
      LOGICAL CONTIN
C                ! continuation flag []
      LOGICAL OK
C                ! ZREPGY call completed ok []
      LOGICAL STOPP
C                ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZREPGY '/,
     1 ONE/1/,
     1 STOPP/.TRUE./
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES DEPTH1 NEEDED ON STACK TO COMPUTE REPGYP'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C........make up a uniform mesh of times based on DELT
         NT = 1 + INT(TLIMIT/DELT)
         DO 5 J = 1,NT
            TIM(J) = DELT * DBLE(J-1)
5        CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........determine values of REPGYP for times passed down
            CALL INTETS(NT,TIM,ONE,VAL,OK)
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values of REPGYP computed
                  DEPTH = VAL(J)
                  IF (DEPTH .LT. DUNSAT) THEN
                     VAL(J) = TFFUNC(TIM(J))
                  ELSE
                     VAL(J) = 0.D0
                  END IF
10             CONTINUE
            END IF
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZSO4MI(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C*************************************************************** ZSO4MI
C
C Calculate SO4MIX from CONCS8, ION, and GYPLFT on stack.
C
C Precondition:  There are at least 3 time series on stack, in order
C                   from the top of stack, GYPLFT, ION, CONCS8.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values for SO4MIX obtained by
C                      application of process 1.2.3.5.
C                END IF
C
C  86-MAY-01  VERSION 01  T. ANDRES
C
C**********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'H:MAXARR.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
      INCLUDE 'TSPACE.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
      DOUBLE PRECISION ZERO
C                ! 0.D0 []
C
      INTEGER J
C                ! general loop index []
      INTEGER NT
C                ! number of times in subseries []
      INTEGER ONE
C                ! 1 []
      INTEGER TWO
```

```
C               ! 2 []
      INTEGER THREE
C               ! 3 []
      INTEGER FOUR
C               ! 4 []
C
      LOGICAL CHKSTK
C               ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C               ! continuation flag []
      LOGICAL OK
C               ! ZSO4MI call completed ok []
      LOGICAL STOPP
C               ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZSO4MI '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 TWO/2/,
     1 THREE/3/,
     1 FOUR/4/,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(THREE)
      IF (.NOT.OK) THEN
         MSG = 'THREE TIME SERIES NEEDED ON STACK TO CALCULATE SO4MIX'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C            WRITE(*,900) MODNAM
C900        FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........Determine initial times of SO4MIX time series.
C...........make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 5 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
5           CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of SO4MIX for times passed down.
C...........First interpolate in time series.  VALS1 = CONCSB;
C...........VALS2 = ION; VALS3 = GYPLFT; VALS4 = CONCCA.
            CALL INTETS(NT,TIM,FOUR,VALS1,OK)
            IF (OK) CALL INTETS(NT,TIM,THREE,VALS2,OK)
            IF (OK) CALL INTETS(NT,TIM,TWO,VALS3,OK)
            IF (OK) CALL INTETS(NT,TIM,ONE,VALS4,OK)
C...........Compute SO4MIX.
            IF (OK) THEN
               DO 10 J = 1,NT
C.................invariant: J-1 values computed.
                  IF ( TIM(J) .LE. TFINAL ) THEN
                     VAL(J) = VALS1(J) + VALS4(J)
                  ELSE IF ( VALS3(J) .GT. ZERO ) THEN
                     VAL(J) = VALS1(J) + VALS2(J)
                  ELSE
                     VAL(J) = VALS1(J)
                  END IF
```

```
10             CONTINUE
            END IF
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
      SUBROUTINE ZSULAK(CALTYP,NT,TIM,VAL,CONTIN,OK)
C
C**************************************************************** ZSULAK
C
C Calculate SULAKE from SO4MIX.
C
C Precondition:  There is at least 1 time series on stack.
C                IF (CALL TYPE (CALTYP) = 'TIMES') THEN
C                   no other preconditions
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   0 <= number of times (NT)
C                   TIM(1:NT) are positive times between 0 and
C                      TLIMIT in increasing order.
C                END IF
C
C Postcondition: OK is .TRUE. unless an error was detected, when
C                   it is .FALSE.
C                IF (CALTYP = 'TIMES') THEN
C                   TIM(1:NT) contains times from 0 to TLIMIT at
C                   intervals of DELT.
C                ELSE IF (CALTYP = 'VALUES') THEN
C                   VAL(1:NT) contains values obtained by applying
C                      processes 4.3.4 and 4.3.3 to SO4MIX on the
C                      stack.
C                END IF
C
C  86-APR-10  VERSION 01  T. ANDRES
C
C***********************************************************************
C
      IMPLICIT LOGICAL(A-Z)
C
      INCLUDE 'CPALL.INC'
      INCLUDE 'SPDEP.INC'
      INCLUDE 'H:TSCONS.INC'
C
      CHARACTER CALTYP*(*)
C                ! call type: 'TIMES' or 'VALUES' []
      CHARACTER MODNAM*6
C                ! module name []
      CHARACTER MSG*53
C                ! error message []
C
      DOUBLE PRECISION BGCON
C                ! background concentration 1/10**1.5 [mol/m**3]
      DOUBLE PRECISION EX
C                ! intermediate exponential value [-]
      DOUBLE PRECISION SULNOW
C                ! current value of SULIN [mol/m**3]
      DOUBLE PRECISION SULPRV
C                ! value of SULIN at previous time step [mol/m**3]
      DOUBLE PRECISION TIM(*)
C                ! times in part of time series [time]
      DOUBLE PRECISION VAL(*)
C                ! values of sum in part of
C                ! time series [quantity/time]
      DOUBLE PRECISION ZERO
C                ! constant 0.D0
C
      INTEGER J
```

```
C                 ! general loop index []
      INTEGER NT
C                 ! number of times in subseries []
      INTEGER ONE
C                 ! 1 []
C
      LOGICAL CHKSTK
C                 ! check that minimum number of ts are in stack [-]
      LOGICAL CONTIN
C                 ! continuation flag []
      LOGICAL OK
C                 ! ZSULAK call completed ok []
      LOGICAL STOPP
C                 ! signal to stop processing []
C
      DATA
     1 MODNAM/'ZSULAK '/,
     1 ONE/1/,
     1 STOPP/.TRUE./,
     1 ZERO/0.D0/
C
C.....check precondition
      OK = CHKSTK(ONE)
      IF (.NOT.OK) THEN
         MSG = 'TIME SERIES SO4MIX NEEDED ON STACK TO COMPUTE SULAKE'
         CALL WRERR(MODNAM,MSG,STOPP)
      END IF
      CONTIN = .TRUE.
      IF (OK) THEN
         IF (CALTYP .EQ. 'TIMES') THEN
C           WRITE(*,900) MODNAM
C900        FORMAT(' ** ',A7,'** CALLED WITH CALTYP = TIMES')
C...........Make up a uniform mesh of times based on DELT
            NT = 1 + INT(TLIMIT/DELT)
            DO 10 J = 1,NT
               TIM(J) = DELT * DBLE(J-1)
10          CONTINUE
         ELSE IF (CALTYP .EQ. 'VALUES') THEN
C...........Determine values of SULAKE for times passed down using
C...........finite difference approach.  First get values for SO4MIX.
            CALL INTETS(NT,TIM,ONE,VAL,OK)
            IF (OK) THEN
C..............Initialize SULNOW and BGCON.
               BGCON = 10.D0**(-1.15D0)
               SULNOW = BGCON
               DO 20 J = 1,NT
C.................invariant: J-1 values of SULAKE computed.  SULNOW
C.................stores the value of SULIN at TIM(J-1).
                  SULPRV = SULNOW
                  SULNOW = VAL(J)*2.5D-2 + BGCON
                  IF (TIM(J) .LE. ZERO) THEN
                     EX     = EXP( - DELT/XLAML )
                     VAL(J) = BGCON*EX + SULNOW*(1.D0-EX)
                  ELSE
                     EX     = EXP( - (TIM(J)-TIM(J-1)) / XLAML )
                     VAL(J) = VAL(J-1)*EX + SULNOW*(1.D0-EX)
                  END IF
20             CONTINUE
            END IF
C...........Make sure this routine is not called again with more times,
C...........as this algorithm is not repeatable.
            CONTIN = .FALSE.
         ELSE
C...........unidentified call type
            MSG = 'UNIDENTIFIABLE CALL TYPE,' //
     1      ' SHOULD BE TIMES OR VALUES'
            CALL WRERR(MODNAM,MSG,STOPP)
            OK = .FALSE.
         END IF
      END IF
      RETURN
      END
```